UNIVERSITÄT LEIPZIG

Lese- und Übungsbuch Datenbanken:

E/R- und Relationenmodell

Dieter Sosna
Institut für Informatik
Abt. Datenbanken

Version 0.9 vom 8. Dezember 2008

Vorwort [1]

In der Arbeit [2] aus dem Jahre 1970 hat *Ted Codd* , der für IBM Research tätig war, das Relationenmodell vorgestellt. Es enthielt ein konzeptionell neues Herangehen an die Datenbankmodellierung.

Vor dem Relationenmodell gab es schon mehrere Versuche, Modelle für die Datenbankarbeit zu entwickeln. So entstanden in den sechziger Jahren des vergangenen Jahrhunderts das *hierarchische Modell* und das *Netzwerkmodell* . Sie wurden in den beiden folgenden Jahrzehnten in Datenbankverwaltungssytemen (DBVS, *engl.* DBMS) implementiert. Nachteilig in beiden Modellen waren die sehr aufwendigen und tiefgreifenden Änderungen in der Datenbank, die notwendig wurden, wenn kleine Veränderungen in der Datenstruktur zu realisieren waren.
Das von *Codd* angeregte Modell zeichnet sich durch Klarheit und Einfachheit aus. *Codd* benutzte die Relation als Datenstruktur, die leicht an verschiedene Anforderungen angepasst werden kann.

Ein anderes Modell, das Entity-Relationship-Modell - kurz E/R-Modell, wurde von Chen ([2]) entwickelt. Chen hat dieses Modell 1976 in einem Aufsatz [1] erstmals vorgestellt und es hat sich sehr schnell weltweit durchgesetzt. Obwohl er es primär für den Datenbankeinsatz entwickelt hat, kann es auch in anderen Umgebungen zur Modellierung verwendet werden. ([3]) Im Vergleich zum Relationenmodell verwendet das E/R-Modell zwei Grundkonzepte, nämlich die von Entitätsmenge und Relationshipmenge ([4]). Viele Modellierungsaufgaben lassen sich mit seiner Hilfe systematisch erledigen. Da zudem ein Regelwerk existiert, wie für Datenbankanwendungen systematisch aus einem E/R-Modell ein dazu passendes Relationenmodell entwickelt werden kann, eignet es sich sehr gut zum logischen Datenbankentwurf, wenn das Ziel der Entwicklung in einer relationalen Datenbank besteht ([5]).
Die Beschreibung des E/R- und des Relationenmodells sind Inhalt des vorliegenden Lese- und Übungsbuches.

Das Lese- und Übungsbuch ist als ergänzende Literatur zu verstehen, es soll auch Ansichten des Themas vermitteln, die in der Vorlesung - meist aus Zeitgründen

[1] Entwurf 1 vom 8. Dezember 2008
[2] Für eine erste Information siehe *http://de.wikipedia.org/wiki/Peter_Chen*
[3] Weitere allgemeine Informationen unter der URL *http://de.wikipedia.org/wiki/Entity-Relationship-Modell*.
[4] Wir verwenden hier bewusst die nicht übersetzten englichen Begriffe, um auszudrücken, dass wir uns auf das E/R-Modell beziehen.
[5] Zum Entwurf objektorientierter Datenbanken ist das UML-Modell vorzuziehen.

- unberücksichtigt bleiben müssen. Keinesfalls soll und kann es den Besuch von Vorlesungen oder Übungen ersetzen.

Dieter Sosna

Leipzig, im November 2008

Inhaltsverzeichnis

1 Modellierung im allgemeinen und in der Informatik 7
- 1.1 Modellbildung . . . 7
  - 1.1.1 Modellbildung . . . 7
  - 1.1.2 Besonderheiten der Modellbildung der Informatik . . . 9
  - 1.1.3 Modellanwendung und Informatik . . . 10

2 Das Entity-Relationship-Modell 13
- 2.1 Entitätsmengen . . . 13
  - 2.1.1 Attribute . . . 13
  - 2.1.2 Entitätsmengen . . . 16
  - 2.1.3 Schlüsselkandidaten, Primärschlüssel . . . 17
  - 2.1.4 Ausschluss von der Schlüsselbildung . . . 18
  - 2.1.5 Graphische Darstellung von Entitätsmengen und Attributen . 19
- 2.2 Relationship-Mengen . . . 22
- 2.3 Problemfälle . . . 25
  - 2.3.1 Schwache Entitätsmengen . . . 25
  - 2.3.2 Kein Schlüsselkandidat? . . . 27
- 2.4 Modellierung weiterer Eigenschaften . . . 28
  - 2.4.1 Grad einer Relationship-Menge . . . 28
  - 2.4.2 Abbildungstypen . . . 29
  - 2.4.3 Kardinalitätsrestriktionen . . . 31
  - 2.4.4 Kardinalitätsrestriktionen bei n-ären Beziehungen . . . 33
- 2.5 Defizite im E/R-Modell . . . 34
- 2.6 Bewertung . . . 34
- 2.7 Ergänzungen / Erweiterungen . . . 35
- 2.8 Beispiel für einen E/R-Entwurf . . . 37

3 Das Relationenmodell 39
- 3.1 Modellbeschreibung . . . 40
  - 3.1.1 Attribute . . . 40
  - 3.1.2 Relationenmengen . . . 40
    - Grundlegende Merkmale - Begriffe . . . 40
    - Notation . . . 42
    - Weitere Merkmale . . . 43
  - 3.1.3 Schlüsselkandidaten, Primärschlüssel . . . 44
- 3.2 E/R - Transformation . . . 45
  - 3.2.1 Fremdschlüssel . . . 46

3.2.2 Umformregeln 47
Attribute, starke Entitätsmengen 47
Relationship-Mengen 48
Schwache Entitätsmengen, mehrfache und zusammengesetzte Attribute 51
Umsetzung der Erweiterungen 54
3.3 Beispiel 57
3.4 Rücktransformation in das E/R-Modell 59
3.5 Bewertung des Relationenmodells 60

4 Anhang 63
4.1 Bankenbeispiel als E/R-Schema 63
4.2 Notation 65
4.3 Bankenbeispiel - Relationenmodell 67
4.4 Eigene Notizen 72

Literaturverzeichnis 73

1 Modellierung im allgemeinen und in der Informatik

Das Ziel dieses Kapitels ist es, einige grundlegende Aussagen über das Wesen der Modellbildung zu treffen und danach auf die Besonderheiten der Informatik einzugehen.

1.1 Modellbildung

1.1.1 Modellbildung

Der Ausgangspunkt ist ein Ausschnitt der realen Welt oder der Begriffswelt unseres Denkens. Er wird i.A. *Miniwelt* genannt. Eine Miniwelt besteht aus Objekten und Beziehungen zwischen diesen Objekten. In der Welt gibt es real existierende Objekte und Objekte, welche ein Produkt des Denkens sind. Auf Objekte und Beziehungen können Aktionen einwirken, die zu Veränderungen der Objekte und Beziehungen führen.

Definition 1.1: Modell
Unter einem Modell verstehen wir ein komplexes Ergebnis des Denkens, welches aus folgenden Komponenten besteht: einer Menge von Modellobjekten, einer Menge von Beziehungen zwischen den Modellobjekten (Modellbeziehungen) und einer Menge von Modelloperationen, die auf die Modellobjekte und -beziehungen anwendbar sind. Das Ergebnis einer solchen Anwendung ist wieder ein Element aus einer der Mengen der Modellobjekte bzw. -beziehungen.([1])([2])

Der Zusammenhang zwischen einer Miniwelt und einem dazugehörigen Modell wird durch eine Wissenschaft vermittelt. Deren Begriffe liefern die semantische Beschreibung der Modellobjekte und Modellbeziehungen. Nur sie erlauben dem Menschen, die Semantik eines Modells, seine Beziehung zur Miniwelt zu verstehen.

[1] Mit Blick auf das spätere Ziel unterscheiden wir zwischen Modellobjekten und Modellbeziehungen. Es ebenso möglich, die Modellbeziehungen als spezielle Objekte im Modell zu betrachten.

[2] Die Eigenschaft, dass die Anwendung einer Modellaktion auf ein Modellobjekt oder eine Modellbeziehung wieder ein solches Objekt bzw. eine solche Beziehung zum Ergebnis hat, ist wesentlich für die mathematische (theoretische) Beschreibung von Modellen. Diese Eigenschaft führt auf den Begriff der Algebra im Sinne einer mathematischen Struktur (s. Wikipedia: URL *http://de.wikipedia.org/wiki/Algebra#Algebra_als_mathematische_Struktur*). Ein Beispiel aus dem Datenbankbereich ist die *Relationenalgebra*, die Gegenstand eines weiteren Bandes dieses Lesebuchs sein wird.

Den Vorgang (Prozess) des Abbildens eines Teilgebiets der Welt nennt man *Modellieren*. Dieser Vorgang wird dadurch charakterisiert, dass in das Ergebnis nicht alle Eigenschaften des zu Modellierenden aufgenommen werden, sondern nur die für einen konkreten Kontext wichtigen. Dies führt i.A. zu einer Vereinfachung des Modells gegenüber dem urprünglichen Sachverhalts und ermöglich dadurch vielfach erst die Anwendungen von Theorien usw.
ıCharakteristisch für die Modellierung ist die Tatsache, dass der Zusammenhang zwischen der Welt und dem Modell idealerweise durch einen Isomorphismus beschrieben wird. Zum Ziel der Modellierung gehört auch das Finden geeigneter Modellobjekte, Modellbeziehungen und Modellaktionen und die Konstruktion des Isomorphismus zur Realität.
Die Modellierung ist ein kreativer Prozess, der das Verstehen der Semantik der Objekte der Miniwelt und der Begriffe der vermittelnden Wissenschaft voraussetzt.

Korrektheit

Die Prüfung der Korrektheit eines Modells beruht im wesentlichen auf der o.g. Forderung, dass das Verhältnis zwischen Miniwelt und Modell durch einen Isomorphismus beschrieben wird. Sie erfolgt durch Versuche, in denen zu Aktionen in der Realität parallel auch die entsprechenden, durch den Isomorphismus bestimmten Modellaktionen auf den entsprechenden Modellobjekten ausgeführt werden und die Ergebnisse unter Einbeziehung des Isomorphismus verglichen werden. Nur wenn die Vergleiche hinreichend oft positiv ausfallen, d.h. realer Versuch und Modellversuch zu isomorphen Ergebnissen führen, gilt die Korrektheit als nachgewiesen, das Modell als *korrekt*.
Dieser Prozess der Prüfung eines Modells bedarf einer sehr wichtigen Anmerkung: Es ist nicht möglich, ein Modell positiv zu evalieren, d.h. seine Korrektheit festzustellen. Ein positiv bewerteter Vergleich zwischen Modellergebnis und entsprechenendem Ergebnis realer Versuche liefert nur keinen Grund, ein Modell als falsch abzulehnen. Die Kunst der Modellevaluierung besteht nun darin, das Modell mit solchen Versuchen zu testen, die auch eventuell problembehaftete Teile des Modells betreffen. Deswegen setzt die die Konzeption der Evaluierung tiefe Kenntnisse der zu modellierenden Miniwelt und des Modells voraus. Sind die zugehörigen Modellrechnungen alle in Übereinstimmung mit den Versuchsergebnissen, kommt die Hoffnung ins Spiel, nämlich die Hoffnung, dass das Modell sich in anderen Fällen ähnlich gut erweist. Dann kann anstatt einer Aktion in der Realität die entsprechende Modellaktion auf Modellobjekten ausgeführt werden. Das Ergebnis - ein Modellobjekt, das mit anderen Modellobjekten durch Modellbeziehungen verbunden ist - lässt sich dann in die Realität zurücktransformieren und liefert ein in der Realität korrektes Ergebnis.
Die Gültigkeit oder Anwendbarkeit eines Modells muss immer wieder kritisch eingeschätzt werden, indem die Ergebnisse von Modellrechnungen immer hinsichtlich ihrer Plausibilität hinterfragt werden. Die Anwendbarkeit eines Modells

kann auf einen Bereich der Realität beschränkt bleiben oder auch später wieder aufgehoben werden, wenn sich neue Erkenntnisse über die Miniwelt ergeben. Dies ist insbesondere im Bereich des wissenschaftlichen Forschung ein häufig zu beobachtender Vorgang. Diesen Vorgang des Ersetzens eines Modells bzw. der Feststellung eines nur eingeschränkten Gültigkeitsbereichs kann an der Entwicklung der verschieden Atommodelle in der Physik, insbesondere im 20. Jahrhundert, sehr gut verfolgt werden.

1.1.2 Besonderheiten der Modellbildung der Informatik

Was ist Informatik?

Definition 1.2: Informatik
Informatik ist die Wissenschaft von der Informationsverarbeitung, der Konstruktion und dem Betrieb der dazu benutzten Maschinen.

Die Informatik ist, verglichen mit gestandenen Wissenschaften wie Mathematik und Physik, eine sehr junge Wissenschaft ([3]), deren Entwicklung noch sehr rasant verläuft. Deshalb ist es schwierig, eine Definition dieses Wissenschaftsgebiets zu geben und die Definition 1.2: möge als Arbeitsdefinition gelten.

Kommentar: In dieser Wissenschaft treffen offenbar zwei Wissenschaftkonzepte zusammen.
Das erste Konzept(... Wissenschaft von der Informationsverarbeitung ...) betont stärker die theoretische Seite: Hier werden Theorien entwickelt, Themen aud einem hohen Abstraktionsniveau behandelt. Dieser Teil der Informatik ist eng mit einigen Bereichen der Mathematik verbunden, die Übergänge sind teilweise fließend.
Im zweiten Konzept (... Wissenschaft von ... der Konstruktion und dem Betrieb der dazu benutzten Maschinen.) stellt sich die Informatik in die Reihe der Ingenieurwissenschaften. Neben eigenen Beiträgen werden hier Mathematik, Physik,

[3] Wikipedia (URL: *http://de.wikipedia.org/wiki/Sexagesimalsystem*) „Erstmalige Nachweise als schriftliches Rechensystem reichen in die Zeit der Sumerer um 3300 v. Chr. zurück."
Der Begriff *Informatik* wurde 1957 von *Karl Steinbruch* in einer Publikation verwendet (Wikipedia: URL *http://de.wikipedia.org/wiki/Informatik*).
Der Beginn der Wissenschaft Informatik ist sicher früher anzusetzen. War die *Shannon*sche Theorie der Information (1948) oder die Festlegung von Algorithmen (z.B. die Patentierung des Soundex-Algorithmus im Jahre 1918 durch *R. C. Russell*) oder waren die noch früheren die Veröffentlichung von Anleitungen zum Rechnen mit mechanischen Rechenmaschinen der Beginn der Informatik? Nicht zu vergessen ist auch das am 8.1.1889 zum Patent angemeldete *Hollerith*-System zur Verarbeitung von Daten auf Lochkartenbasis.
Diese Frage sollen hier nur aufgeworfen werden.

andere Naturwissenschaften und auch andere Ingenieurwissenschaften genutzt, um die für die Informationsverarbeitung genutzten Maschinen zu beschreiben, die Technologie ihrer Herstellung und Verfahren für ihre Anwendung zu untersuchen und weiterzuentwickeln.
Beide Konzepte sind an vielen Punkten verbunden.

Von der Informatik selbst sind ihre Anwendungen in anderen Wissenschaftsgebieten zu unterscheiden. Die hier oft vorhandene unklare Begriffsbildung führt andernfalls zu merkwürdigen Ergebnissen: so glaubt jemand häufig, Informatikkenntnisse zu haben, nur weil er ein Textverarbeitungsprogramm nutzt und dort Makros neu erstellt. Niemand käme auf den Gedanken, ein Hochfrequenzingenieur zu sein, nur weil er ein Radio bedienen kann.
Hinsichtlich der Einschätzung der Tätigkeit des Programmierens sei auf *Weizenbaum* [8] verwiesen. Programmieren kann mit einem Handwerk verglichen werden, welches der Informatiker bis zu einem gewissen Grade beherrschen sollte, aber wie bei jedem Handwerk gibt es auch hier den Fachmann, den Programmierer. Der wiederum hat Programmier-Aufträge in der vorgegebenen Zeit in einer überprüfbaren technischen Qualität zu erledigen.

1.1.3 Modellanwendung und Informatik

Typisch für die Anwendung der Informatik ist nun ein zweiter, gleichartiger Modellierungsschritt. Jetzt fungiert das Modell als Ausgangspunkt und die Rolle des Modells übernimmt eine rechnerinterne Darstellung (RID), in dem Modellobjekte und -beziehungen einerseits und die Modellaktionen andererseits jeweils Folgen von Zeichen (Zeichenketten) über dem Alphabet {0,1} sind, die nach bestimmten Regeln konstruiert werden und in dem das Ergebnis einer Anwendung einer Zeichenkette aus der Menge der Modellaktionen auf die Zeichenketten der Modellobjekte und -relationen definiert wird. Die Modellierung wird durch Hardware, Betriebssystem, Programmiersprachen und solche Sprachen wie SQL vermittelt. Die Modellverifikation heißt hier Programmtesten. Charakteristisch für diesen zweiten Schritt ist, dass das Modell i.A. formal definiert ist, die Verarbeitung kann durch eine Maschine erfolgen.

Im Sinne dieser Definition gehört die Anwendung eines Modells zu den Anwendungen der Informatik. Das heißt aber auch, dass bei der Anwendung die Begriffswelt des Anwendungsgebiets und die Semantik der dort verwendeten Begriffe eine wichtige Rolle spielt. Hier wird also die Informatik durch Vermittlung einer anderen Wissenschaft wirksam. Besonders gut lässt sich dies im Bereich der Datenintegration nachweisen. Hier zeichnet sich als eine Hauptaufgabe die Benutzung und die Verarbeitung der Begriffswelt einer oder mehrerer Wissenschaften ab.
Für viele praktisch ausgeführte Modelle muss jedoch festgestellt werden, dass ins-

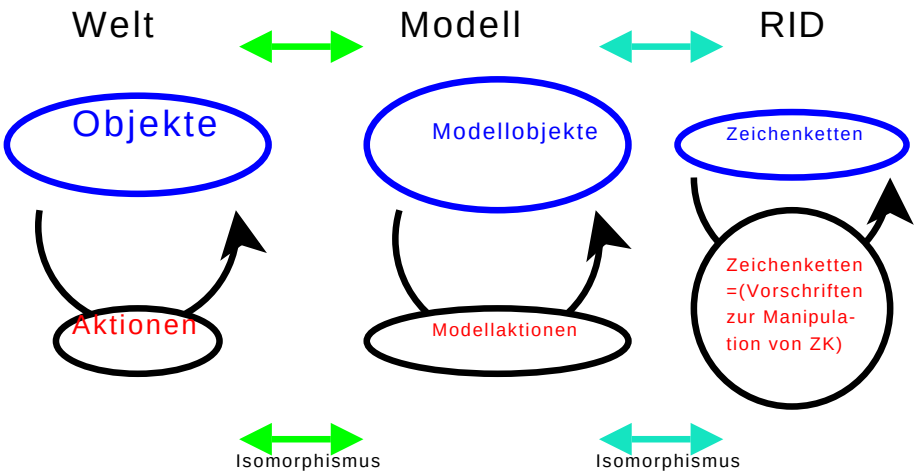


Abbildung 1.1: Modellierung in der Informatik

besondere dann, wenn nur mit der zweiten Stufe des Modells gearbeitet wurde bzw. wenn für die Arbeit nur die zweite Stufe zur Verfügung steht, der konkrete Bezug zur Begriffswelt häufig verloren geht bzw. nicht mehr sicher erkennbar ist. Deshalb soll nochmal ausdrücklich festgestellt werden, dass zum Modell, das nach dem ersten Modellierungsschritt festgelegt ist, auch die Beschreibung des Isomorphismus, also des Zusammenhangs mit der Miniwelt gehört.([4]) Der zweite Modellierungsschritt ergibt sich aus den in der Informatikdefinition enthaltenen ingenieurwissenschaftlichen Komponente: Algorithmen, Datenstrukturen sowie deren Kodierung in einer Maschine zum Zweck der Informationsverarbeitung.

Die Modellierung, die bei der Erstellung einer Datenbank zu leisten ist, fällt in den ersten Schritt der Modellbildung und nur zum Teil in den zweiten. Zum ersten Schritt gehören das Erarbeiten eines semantischen Modells in Form eines E/R- oder eines UML-Modells, ggf. auch die Transformation in ein relationales Modell. Das Erstellen der zugehörigen SQL-Anweisungen stellt den Übergang zum zweiten Schritt dar.

[4] Diese Forderung wird in den Definitionen 2.2: (Seite 13) und 2.7:(Seite 16) aufgegriffen.

2 Das Entity-Relationship-Modell

In diesem Modell gibt es zwei Grundelemente: Entitätsmengen und Relationship-Mengen ([1]).
Das Entity-Relationship-Modell, kurz E/R-Modell genannt, enthält die Möglichkeit, seine Bestandteile durch Symbole graphisch darzustellen. Diese Darstellungen sind beim Entwurf sehr hilfreich, da sie wesentliche Zusammenhänge in kompakter, übersichlicher Form beschreiben (siehe S. 19).

2.1 Entitätsmengen

Die Entitätsmengen haben als Elemente Entitäten.
Eine Entität ist im Sinne der gegebenen Modelldefinition ein Modellobjekt, mit anderen Worten:

Definition 2.1: *Eine* **Entität** *ist das Modellobjekt zu einem Objekt der zu modellierenden Welt.*

Da die zu modellierende Welt nie alles umfassen kann, also stets einen (kleinen) Ausschnitt aus der realen Welt oder / und einer gedanklichen Konstruktion darstellt, wird für diesen zu modellierenden Ausschnitt der Begriff **Miniwelt** verwendet.

2.1.1 Attribute

Alle Entitäten haben im E/R-Modell (nur) eine gemeinsame Eigenschaft: **Entitäten haben in dem Modell nichts als Attribute**. Dies hat eine zentrale Bedeutung des Attributbegriffs zur Folge. Alle weiteren Aussagen über diese Modellobjekte müssen als Aussagen über die Attribute formuliert werden. Ein Attribut beschreibt im Modell eine Eigenschaft des Objekts.

Definition 2.2: Attribut
Ein Attribut ist eine Komponente einer Entität. Ein Attribut wird beschrieben durch folgende Angaben:

1. **Name.** *Der Name eines Attributs ist sein Identifikator. Er muss innerhalb der Attribute der Entität eindeutig sein.*

[1] Bei diesem Begriff wird absichtlich auf den Versuch eine Übersetzung verzichtet, um Verwechslungen mit später zu besprechenden Bildungen zu vermeiden. Somit wird dieser Begriff fest an dieses Modell gebunden und auch nur in diesem verwendet.

2. **Wertebereich.** *Der Wertebereich ist die Menge aller Attributwerte, die in der zu beschreibenden Miniwelt für die durch das Attribut modellierte Eigenschaft auftreten können. Für jede Entität wird das Attribut mit einem Wert aus dem Wertebereich belegt.*
3. **Kontext.** *Der Kontext erklärt, zu welcher Entität das Attribut gehört.*
4. **Semantik.** *Die Semantik eines Attributs erklärt die Bedeutung des Attributs im Kontext der zu modellierenden Realität.*

Die Beschreibung der Semantik erfordert grundsätzlich Mittel, die das Modell selbst nicht bereitstellt, also eine *Metasprache*.
Der einfachste Weg ist die **textuelle Beschreibung der Semantik** mit Hilfe eines Fachwortschatzes einer Wissenschaft, deren Begriffe zur Modellbildung verwendet wurden (und der üblichen Sprache).
Als **Nachteil** dieses einfachen Vorgehens sind anzumerken:
- das Vokabular wird nicht (umfassend) kontrolliert, es ist also von den Benutzern beliebig erweiter- und ergänzbar, wobei keine Qualitätssicherung möglich ist.
- die Beschreibung ist nicht automatisch verarbeitbar.

Natürlich hat ein solches Vorgehen auch **Vorteile:**
- durch Menschen lesbar,
- leichte Erweiterbarkeit des Wortschatzes zur Beschreibung der Semantik im Falle der Weiterentwicklung der Wissenschaft, die die Modellierung vermittelt.

Das Erkennen / Kennen der Semantik der Komponenten ist die Voraussetzung jeder sinnvollen Verwendung des Modells.

Beispiel 2.3: Attributbeschreibung
Das Attribut „Farbe“ welches die hauptsächlich an der Charosserie vorkommende Farbe sein soll, hat bei einem fiktiven Autotyp X die Werte „schwarz“, „tornadorot“, „divingblue“ und „bambusgrey“.

Das Beispiel einer Attributbeschreibung erklärt sich nahezu von selbst. Der Name ist „Farbe“, der Wertevorrat umfasst die vier angegebenen Zeichenketten, in dem im Beispiel beschriebenen Zusammenhang sind andere Werte für das Attribut „Farbe“ nicht erlaubt. Die anderen Angaben gehören zur Semantik (...die hauptsächlich an der Karosserie vorkommende Farbe ...) bzw. zum Kontext (... bei einem fiktiven Autotyp X ...).

Definition 2.4: Zwei Attribute sind (semantisch) gleich *genau dann, wenn sie in den Angaben nach Definition 2.2: mit Ausnahme von Namen und Kontext übereinstimmen.*

Diese Definition beinhaltet ein Problem. Sie ist für Menschen verständlich und auch praktikabel. Soll jedoch der Vergleich automatisch erfolgen, setzt dies eine Form der

Angaben voraus, die durch Maschinen verarbeitet werden kann, diese Forderung ist für Punkte 4. der Attributbeschreibung i.a. nicht erfüllt.

Die Wahl und die Beschreibung der Attribute ist eine schöpferische Leistung im Modellierungsprozess. Die Qualität des Modells kann davon abhängen. Dies wird im Zusammenhang mit der Bildung von Entitätsmengen genauer diskutiert (siehe Abschnitt 2.1.2, S. 16)

Es soll noch ausdrücklich angemerkt werden, dass Attribute durchaus komplexer Natur sein können. Dazu gehören sowohl aus Komponenten zusammengesetzte Attribute als auch mehrfache Attribute. So können der Name einer Person aus dem Titel, Vornamen, dem Nachnamen und Namenszusätzen bestehen. Titel und Vornamen selbst können mehrfache Attribute darstellen, genannt seien Wissenschaftler mit mehreren Doktortiteln oder Personen mit mehreren Vornamen. Der Wertevorrat eines solchen komplexen Attributs besteht dann aus dem entsprechenden Kreuzprodukt der Wertevorräte der Komponenten.

Definition 2.5: Einfache, mehrfache und zusammensetzte Attribute
Ein Attribut, welches für eine Entität mehrere Werte annehmen kann, heißt **mehrfaches Attribut**.
Ein Attribut, welches sich semantisch in mehrere Komponenten aufteilen lässt, welche eine eigene Semantik im Rahmen der betrachteten Miniwelt haben und dessen Komponenten auch einzeln im Modell dargestellt werden, heißt **zusammengesetztes Attribut**.
Die Attribute, die weder zusammengesetzt noch mehrfach sind, heißen **einfach**.

An dieser Definition, besonders an der Festlegung zusammengesetzter Attribute wird deutlich, wie das Verständnis des Modellierers für die Semantik in das Ergebnis des Modellierungsprozesses eingehen kann. Sein Verständnis entscheidet häufig darüber, ob ein Attribut durch Komponenten feiner granuliert dargestellt wird. So kann eine *Adresse* als einfaches Attribut behandelt werden oder in Komponenten unterteilt werden: *Adresse* = (*Postleitzahl, Ortsname, Straßenname, Hausnummer, Wohnungsnummer*).

Mit Hilfe der Attribute können die Entitäten beschrieben werden:

Axiom:
Eine Entität wird in E/R-Modell durch die Angabe der dieser Entität zugeordneten Attributnamen-Wert-Paare in ausreichender Weise gekennzeichnet.

Wird die Gültigkeit dieses Axioms in Zweifel gezogen, verliert das Modell seine Grundlage. **Das Finden der Attribute und die Beschreibung der Entitäten durch die Attribut-Wert-Paare bilden den ersten Schritt der Modellierung.**

2.1.2 Entitätsmengen

Im folgenden Schritt wird die Menge aller Entitäten strukturiert, indem Teilmengen gebildet werden.

Definition 2.6: *Zwei Entitäten gehören der gleichen Teilmenge der Menge aller Entitäten an, wenn sie in den Mengen ihrer Attribute (nicht der Werte) übereinstimmen. Jede der so gebildeten Teilmengen heißt* **Entitätsmenge.**

Auf Grund dieser Definition wird jede Entität jeweils genau einer Teilmenge zugeordnet, es findet also eine Zerlegung der Menge aller Entitäten in Klassen ([2]) statt, die hier Entitätsmengen heißen.

Nachdem die Attribute festgelegt sind, ist diese Klassenbildung scheinbar ein formaler Schritt. Die Bildung der Entitätsmenge ist jedoch mehr als nur die Zusammenfassung formal gleichartiger Entitäten. Zunächst ist sie abhängig von der Wahl der Attribute der Entitäten, d.h. die Wahl der Attribute darf nicht losgelöst von den Begriffen in der Miniwelt erfolgen. Es ist wesentlich, dass für Entitäten, die zu einem gemeinsamen Oberbegriff in der Miniwelt gehören, auch Attribute gefunden werden, die in ein gemeinsames Konzept passen. Es ist dann zu prüfen, ob die Zusammenfassungen im Rahmen der Begriffswelt, die der Miniwelt zu Grunde liegt, eine eigene Semantik besitzen, ob also eine Entitätsmenge einem Begriff der zu modellierenden Welt entspricht. Die Entitätsmengen bilden eine Grundstruktur des E/R-Modells. Da häufig auf sie zugegriffen wird, erhalten sie jeweils einen eigenen Namen. Und dabei ist mit Blick auf das Endziel der Modellierung auch zu prüfen, ob nicht auch noch generalisiert werden soll.
In der Praxis ist häufig zu beobachten, dass das Finden von Attributen und Entitätsmengen und ihre Zuordnung zu den Konzepten der Miniwelt auf iterative Wiederholungen dieser beiden Schritte führt, wobei auch Umwidmungen und Neuschöpfungen auftreten. Das Finden der richtigen Entitätsmengen und der richtigen Attribute ist ein schöpferischer Prozess, dessen Güte die Eignung des Modells direkt beeinflusst. In diesem Sinne ist die gesamte Modellierung ein krativer Prozess, dessen Ergebnis auch von der Erfahrung und dem Verstehen der Beteiligten (Personen) abhängt.

Definition 2.7: *Die* **Beschreibung einer Entitätsmenge** *erfolgt durch folgende Angaben:*

1. **Name.** *Der Name einer Entitätsmenge gilt als ihr Identifikator. Er muss also eindeutig sein - zumindest als Name einer Entitätsmenge im Kontext des gerade erstellten Modells.*

[2] Der Begriff **Klasse** wird hier im mathematischen Sinn mit der Bedeutung einer vollständigen und disjunkten Einteilung in Teilmengen benutzt.

2. **Menge der Attribute**, *die für alle Elemente relevant sind.*
3. **Semantik**, *ausgedrückt in der Begriffswelt der zu modellierenden Miniwelt.*
4. *Beschreibung* **Miniwelt**, *in der die Entitätsmenge auftritt.*

2.1.3 Schlüsselkandidaten, Primärschlüssel

Da die Teilmengenbildung den mathematischen Mengenbegriff unterstellt, nach dem es in einer Menge keine Duplikate gibt, ergeben sich für die Modellierung Konsequenzen. So ist durch die Wahl bzw. Festlegung der Attribute dafür zu sorgen, dass Objekte der Miniwelt, die als verschieden anzusehen sind, auch auf unterschiedliche Entitäten abgebildet werden. Nach der Bildung der Entitätsmengen gilt für jede Entitätsmenge, dass sie eine Menge im Sinn des mathematischen Begriffs ist. Da alle Elemente einer Entitätsmenge nach Konstruktion dieser Mengen genau die gleichen Attribute besitzen, die für jede Entität mit Werten belegt werden, kann die Mengeneigenschaft nur erfüllt werden, wenn es in einer Entitätsmenge keine zwei Entitäten gibt, die in allen Attributwerten übereinstimmen.
Dies kann man auch anders ausdrücken:

Satz 2.8: *Durch Angabe der Werte der (aller) Attribute kann eine Entität innerhalb der zugehörigen Entitätsmenge eindeutig bestimmt werden.*

Schlüsselkandidaten
Diese Aussage motiviert die folgende Frage: Reichen evt. weniger als alle Attribute, um aus den Werten für diese Attribute eine Entität eindeutig zu identifizieren? Die Frage muss dabei so verstanden werden, dass die Antwort nicht nur für die derzeit vorhandenen Entitäten gilt, sondern für alle Entitäten, die auf grund der Wertevorräte der Attribute in der betrachteten Miniwelt überhaupt möglich sind.
Diese Frage kann i.A. nicht formal durch Untersuchung der Wertevorräte entschieden werden, sondern nur durch Betrachtung der Semantik. Dabei sind sowohl die Attribute als auch die Entitätsmengen und ihre Urbilder in der Miniwelt zu betrachten. In praktischen Beispielen ist meist die Beschreibung der Miniwelt um allgemeines Wissen um das modellierte Gebiet zu ergänzen, damit diese Frage beantwortet werden kann.

Definition 2.9: Schlüsselkandidat
Eine Teilmenge $\mathcal{S}$ der Menge aller Attribute eines Entitätsmenge heißt Schlüsselkandidat g.d.w. gilt:

1. *Eine Entität wird durch die Attributwerte des Attribute von $\mathcal{S}$ eindeutig bestimmt, d.h. seien E eine Entitätsmenge, e_1,$e_2 \in E$ mit der Eigenschaft $e_1 \neq e_2$ so folgt $e_1|\mathcal{S} \neq e_2|\mathcal{S}$ ([3])*

[3] Mit $e|\mathcal{S}$ beschreiben wir die Einschränkung von e auf die auf Attribute von $\mathcal{S}$, d.h. es werden nur die Werte dieser Attribute betrachtet.

2. *Aus $\mathcal{S}$ kann kein Element weggelassen werden ohne dass die Eigenschaft (1) verloren geht.*

Die zweite Eigenschaft wird mit dem Begriff **Minimalität des Schlüsselkandidaten** beschrieben. Sie erweist sich als wesentliche Voraussetzung für die Begriffsbildungen der Normalformenlehre ([4]).

Da es, wie oben dargelegt, in einer Entitätsmenge keine zwei Elemente geben kann, die in allen Attributen übereinstimmen, muss es in einer Entitätsmenge mindestens eine Attributkombination geben , die die Bedingung 1. erfüllt. Sie enthält im ungünstigsten Fall alle Attribute. Somit ist die Existenz eines Schlüsselkandidaten (theoretisch) gesichert.

Primärschlüssel
Eventuell gibt es zu einer Entitätsmenge mehrere Schlüsselkandidaten. Aus allen Schlüsselkandidaten wird einer ausgewählt und zum **Primärschlüssel** erklärt. Aus (theoretischer,) die Funktionalität betrachtender Sicht sind dabei alle Schlüsselkandidaten gleichwertig und die Auswahl könnte nach gusto erfolgen. Wie sich noch zeigen wird, macht es aus praktischen Überlegungen heraus Sinn, möglichst einen Primärschlüssel zu wählen, der wenige Attribute enthält, am besten sind deshalb in der Praxis einzelne Attribute geeignet, wenn sie denn die Bedingung (1) für Schlüsselkandidaten erfüllen, um (2) braucht man sich in diesem Fall nicht zu kümmern.

Es wird festgelegt, dass zur Bestimmung einer Entitätsmenge die Feststellung des Primärschlüssels gehört.

In der Praxis kann es vorkommen, dass entgegen aller Theorie praktisch kein Schlüsselkandidat gefunden wird. Die Diskussion dieses Falles wird noch etwas verschoben (siehe S. 27).

2.1.4 Ausschluss von der Schlüsselbildung

Unbestimmte Werte eines Attributs
Bei der Festlegung des Wertevorrats eines Attributs bestehen kaum Einschränkungen. Dies bedeutet, das zum Wertevorrat auch Werte mit einer Semantik gehören können, die eine Unbestimmtheit oder Unsicherheit zu beschreibt. Beispiele dafür sind „Wert unbestimmt“ oder „Wert unbekannt“ oder „Existenz des Wertes unsicher“ oder ... . Träte ein solcher Wert in einem Schlüsselkandidaten

[4] Es ist vorgesehen, die Normalformenlehre in einem weiteren Lesebuch darzustellen.

auf, bedeutete dies, dass die eindeutige Bestimmtheit einer Entität durch den Schlüsselkandidaten verletzt wird. Da jedoch bei der allgemeinen Betrachtung beim Entwurf des Modells nicht sicher ausgeschlossen werden kann, dass nicht irgendwann später eine Entität mit solchen Werten auftritt, wenn diese in der Attributdefinition zugelassen werden,

... wird zusätzlich verboten, dass Attribute mit Werten, die eine Unbestimmtheit ausdrücken, Bestandteil eines Schlüsselkandidaten sein können.

Diese Eigenschaft eines Schlüsselkandidaten wird auch als **Gegenstandsintegrität** bezeichnet ([5]).

Nichteinfache Attribute
Nichteinfache Attribute sind zusammengesetzte oder mehrfache Attribute, in beiden Fällen ist ein Gleichkeitsbegriff definiert: zusammengesetzte Attribute sind gleich wenn sie sie die gleiche Komponentenstruktur haben und komponentenweise gleich sind, mehrfache Attribute sind gleich, wenn sie als Mengen gleich sind (also die gleichen Elemente enthalten). Andernfalls sind sie ungleich. Damit wäre es theoretisch möglich, dass im E/R-Modell zusammengesetzte oder mehrfache Attribute Bestandteile eines Schlüsselkandidaten sind.
Dennoch

sollten nichteinfache Attribute als Bestandteile eines Primärschlüsels vermieden werden.

Der Grund für diese Empfehlung liegt in dem (noch zu besprechenden) Relationenmodell, in welchem nur einfache Attribute existieren. Das Beachten dieser Empfehlung sichert, dass Primärschlüssel aus dem E/R-Modell einfach übernommen werden können. Andererseits kann es vorkommen, dass in jedem Schlüsselkandidaten ein nichteinfaches Attribut auftritt und folglich bei Beachtung der Empfehlung kein Primärschlüssel bildbar ist (vgl. hierzu Abschnitte 2.3.2, Seite 27, und 3.2.2,Seite 47).

2.1.5 Graphische Darstellung von Entitätsmengen und Attributen

Das E/R-Modell kommt mit einer sehr suggestiven graphischen Darstellungsmöglichkeit einher, die in einer ersten Form schon von Chen (1976) eingeführt wurde und in wesentlichen Zügen hier genutzt werden soll ([6]). Diese sogenannten E/R-Diagramme ermöglichen, dass ein Betrachter sich schnell einen Überblick über eine konkrete Modellierung verschafft. Für die verschiedenen Komponenten

[5] Siehe *http://www.gitta.info/LogicModelin/de/html/DBIntegrity_KeyInteg.html*
[6] Eine Übersicht über weitere Notationsansätze findet sich in der freien Enzyklopädie Wikipedia URL: *http://de.wikipedia.org/wiki/Entity-Relationship-Modell#ER-Diagramme*

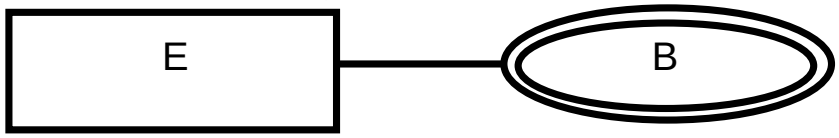
des Modells existieren symbolische Darstellungen.

Festlegung:
Das **Symbol für eine Entitätsmenge** besteht aus einem Rechteck mit einem einfachen Rand, in dessem Inneren die Angaben zur Entitätsmenge entsprechen den Ausführungen im Abschnitt 2.7: (S. 16) stehen.

Um eine gute übersichtlichkeit auch bei großen Darstellungen zu erhalten, wird i.A. jedoch nur der Name der Entitätsmenge eingetragen, die anderen Angaben werden in Listen außerhalb der Diagramme verwaltet. Die Beschränkung auf die Namen gilt i.a. für alle Komponenten dieser Diagramme.

Festlegung:
Die **Attribute** werden durch Kreise (Ellipsen) repräsentiert, die durch eine Strecke mit dem Rechteck ihrer Entitätsmenge verbunden sind. Im Innern der Ellipse befindet sich der Attributname.

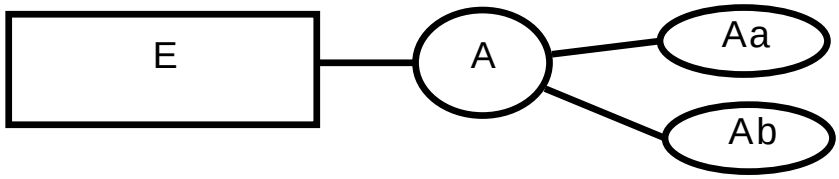
Entitätsmenge mit Attribut



Entitätsmenge E mit mehrfachem Attribut B

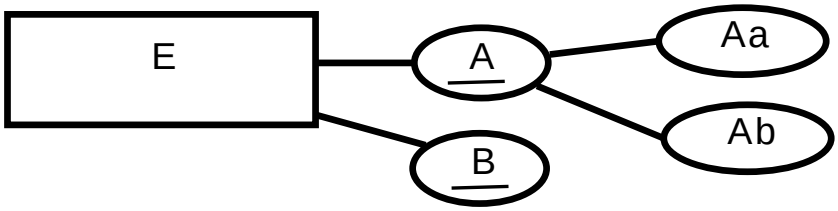
Bei **mehrfachen Attributen** beseht der Rand aus einer Doppellinie.

Bei **zusammengesetzten Attributen** werden an die Attributkreise weitere Kreise für die Komponenten mittels Strecken angehängt .



Entitätsmenge E mit zusammengesetztem Attribut A,
bestehend aus den Komponenten Aa und Ab

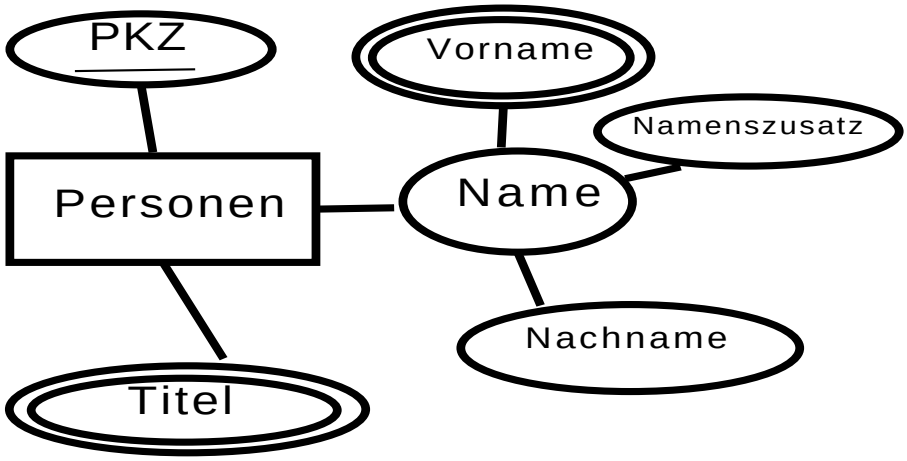
Festlegung:
Die Attribute, die den **Primärschlüssel** bilden, werden **unterstrichen**.



Entitätsmenge E mit zusammengesetztem Primärschlüssel (A,B)

Enthält der Primärschlüssel auch ein zusammengesetztes Attribut, bereitet dies in diesem Modell keinerlei Probleme, die Einzigkeitsforderung aus der Schlüsseldefinition ist auf die Komponenten des zusammengesetzten Attributs sinnfällig anzuwenden (vgl. Abschnitt 2.1.4, Seite 19)

Beispiel 2.10: *Das folgende Bild zeigt die Entitätsmenge* Personen *mit dem Attributen wie auf Seite 15 beschrieben. So ist der* Name *einer Person in der Regel zusammengesetzt, er könnte aus* Titel, Vorname, Namenszusatz *(von, zum, ...) und dem* Nachnamen *bestehen. Titel und Vorname selbst können mehrfache Attribute darstellen, genannt seien Wissenschaftler mit mehreren Doktortiteln oder Personen mit mehreren Vornamen. Für eine problemlose Primärschlüsseldefinition wird die Existenz einer universellen Personenkennzahl* PKZ *vorausgesetzt.*



Entitätsmenge Personen

2.2 Relationship-Mengen

In einem Modell wird es fast immer mehrere Entitätsmengen geben. Zwischen den Elementen zweier oder mehrerer dieser Mengen bestehen im allgemeinen Beziehungen.

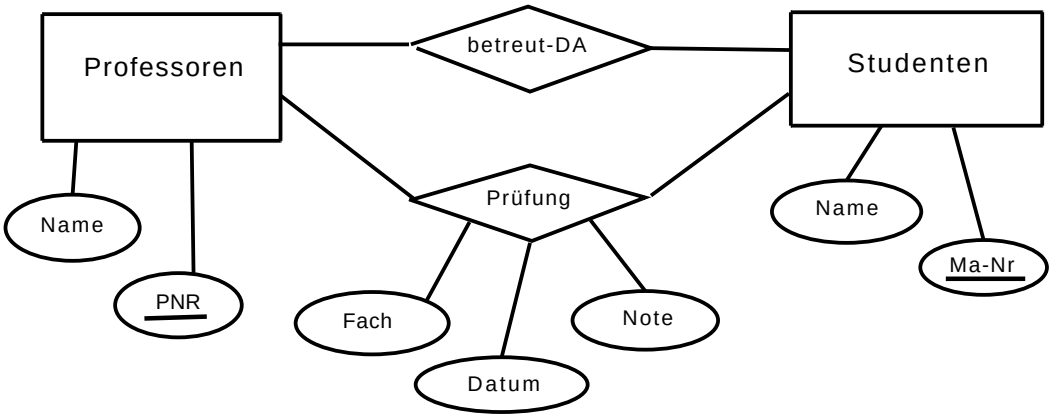
Beispiel 2.11: *In der Miniwelt einer Universität gibts Professoren und Studenten. Beide haben als Attribut einen Namen, ein Professor eine Personalnummer, Studenten eine Matrikelnummer. Die Nummern bilden jeweils den Primärschlüssel. Sinnvollerweise bilden Professoren und Studenten jeweils eine Entitätsmenge. Zwischen einzelnen Studenten und einzelnen Professoren gibt es Beziehungen: Professor P1 betreut die Diplomarbeiten der Studenten S1,S2. Professor P2 prüft die Studenten S1, S3, S4. Bei einer Prüfung muss natürlich das Fach festgelegt werden, dann ist das Prüfungsdatum und die Note festzuhalten. Diese Daten bilden Attribute der Beziehung.*

Definition 2.12: Relationship-Mengen *Die konkreten Beziehungen zwischen den Entitäten von Entitätsmengen werden zu Mengen gleichartiger Beziehungen zusammengefasst, diese Mengen heißen Relationship-Mengen. Die Gleichartigkeit wird durch die gleichen beteiligten Entitätsmengen und gleiche Attribute induziert. Festgestellt wird sie dann durch die gleiche Semantik.*

In unserer Miniwelt entstehen die Relationship-Mengen „betreut-DA“ und “Prüfung“, letztere hat noch die zusätzliche Informationen, die hier *Attribute der Relation* heißen. Diese Attribute werden wie bei den Entitätsmengen notiert.

Festlegung:
Im Diagramm wird eine Relationship-Menge durch einen Rhombus dargestellt.

Die folgende Abbildung zeigt die Idee der Darstellung (für ein komplettes Diagramm müssen noch einige Angaben hinzugefügt werden):



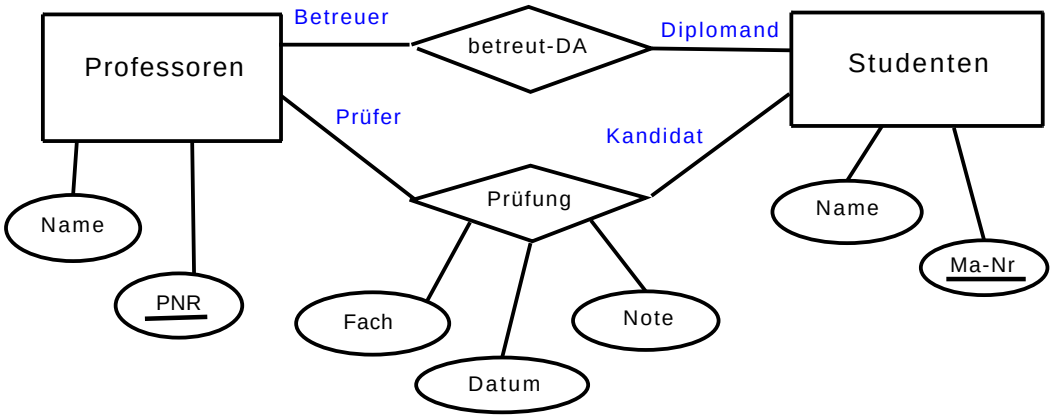
Relationship-Mengen (unvollständig ([7]))

Satz 2.13: *Mathematisch sind die Relationship-Mengen Relationenmengen, also Teilmengen des Kreuzprodukts der Entitätsmengen. Bei zwei durch eine Relationship-Menge verbundenen Entitätsmengen besteht die Beziehung (, wenn sie keine eigenen Attribute hat,) aus Paaren von Entitäten.*

Weitere Angaben im E/R-Diagramm

Im Diagramm kann zusätzlich die Rolle angegeben werden, die die Entitäten in der Beziehung realisieren. Im Bild erscheint nur der Rollenname, die Beschreibung der Rolle, ihre Funktion usw. muss in einer Legende erfolgen. Die Beschreibung wird sicher entfallen können, wenn die Bedeutung des Rollennamens allgemeines Wissen ist.

[7] Dieses und die folgenden Diagramme sind insofern unvollständig, als dass die sog. Abbildungstypen (vgl. Seite 29) und die Kardinalitätsrestriktionen (vgl. Seite 31) noch fehlen. Beide Angaben modellieren Aspekte der Semantik und sollten deshalb angegeben sein.



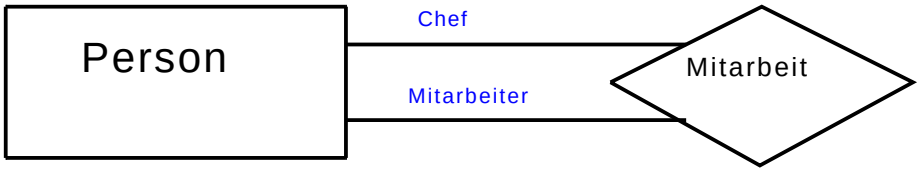
Relationship-Mengen mit Rollennamen (unvollständig)

Es stellt kein Problem dar, wenn eine Relationship-Menge eine Beziehung einer Entitätsmenge mit sich selbst beschreibt. Solche Beziehungen werden auch **reflexive Beziehungen** genannt. Typische Beispiele sind:

a) Entitätsmenge: Staaten (der Erde), Relationship-Menge: Staat 1 grenzt an Staat 2.

b) Entitätsmenge: Personen; Relationship-Menge: Person 1 ist Chef von Person 2 (und umgekehrt ist Person 2 Mitarbeiter von Person 1).

c) Entitätsmenge: Personen; Relationship-Menge: Person 1 ist verheiratet mit Person 2. Hier können (wenn man sich auf Deutschland beschränkt) noch die Rollen *Ehefrau* und *Ehemann* angegeben werden, und es gilt die Nebenbedingung, dass Polygamie verboten ist.



Reflexive Beziehung nach b) (unvollständig)

Bei reflexiven Beziehungen sind Rollennamen für das Verständnis i.a. unverzichtbar. Beim Übergang zum Relationenmodell (siehe S. 39) können sie unter bestimmten Bedingungen die Grundlage für neue Attributnamen sein, mit denen dann die Rolle gekennzeichnet wird.

2.3 Problemfälle

2.3.1 Schwache Entitätsmengen

In diesem Abschnitt sollen Probleme gelöst werden, die entstehen, wenn die Primärschlüssel von Entitätsmengen nicht feststellbar sind. Zur Illustration diene zunächst die folgende Miniwelt:

Aufgabe 2.14: *In einer Firma soll für die Kinder von Mitarbeitern ein Betriebskindergarten eingerichtet werden. Von einen Mitarbeiter seien sein Nachname und die Personalnummer (PNR, Primärschlüssel) bekannt, von den Kindern der Vorname, das Geburtsdatum und ein Lieblingsessen ([8]). Und natürlich gibt es eine Beziehung jedes Kindes zu einem Mitarbeiter (, nämlich dessen Kind zu sein).*

Lösung:
Folgende Feststellung vorab: der E/R-Entwurf zu einer Miniweltbeschreibung ist nicht eindeutig, meist gibt es mehrere Lösungen, so auch hier.

Die **erste Lösung** geht davon aus, dass die Kinder als eigenständige Objekte der Miniwelt angesehen werden sollen.
Bei dieser Annahme können die Vorgaben mit zwei Entitätsmengen „Mitarbeiter“ und „Kinder“ sowie der Relationship-Menge „ist-Kind“ beschrieben werden. Jetzt besteht das Problem, dass die vorgelegten Attribute der Entitätsmenge „Kinder“ nicht geeignet sind, einen Primärschlüssel zu finden. Es ist leicht möglich, dass in dem Kindergarten zwei Kinder mit gleichem Vornamen und gleichen Geburtsdatum existieren. Nach den oben getroffenen Festlegungen kann deshalb „Kinder“ keine Entitätsmenge sein.
Nun ist es jedoch so, dass durch die Beziehung zu den Mitarbeitern die Eindeutigkeit erreicht wird. Die Kinder eines Mitarbeiters haben unterschiedliche Vornamen ([9]). D.h. der Vorname spielt in Verbindung mit der Beziehung zur Entitätsmenge „Mitarbeiter“ die Rolle eines Primärschlüssels, er wird jetzt **Schlüsselergänzung** genannt.

Definition 2.15: *Solche Entitätsmengen, die zur Bestimmung eines Primärschlüssels den Bezug zu einer anderen Entitätsmenge benötigen, heißen im E/R-Modell* **schwache Entitätsmengen**.

In der Diagrammdarstellung schwacher Entitätsmengen wird die Schlüsslergänzung wie ein Primärschlüssel unterstrichen, der besondere Status der

[8] Es ist nicht die Aufgabe des Datenbankentwurfs, ungewöhnliche Attributkombinationen zu bewerten - sie werden bestmöglich realisiert.
[9] Wir sehen hier von den Sonderfällen ab, die durch Scheidung, Wiederheirat u.a. entstehen.

Entitätsmenge durch einen doppelten Rahmen markiert. Wenn die schwache Entitätsmenge Beziehungen zu mehreren anderen Entitätsmengen hat, wird auch diejenige Relationship-Menge doppelt umrandet, die die Eindeutigkeit vermittelt.

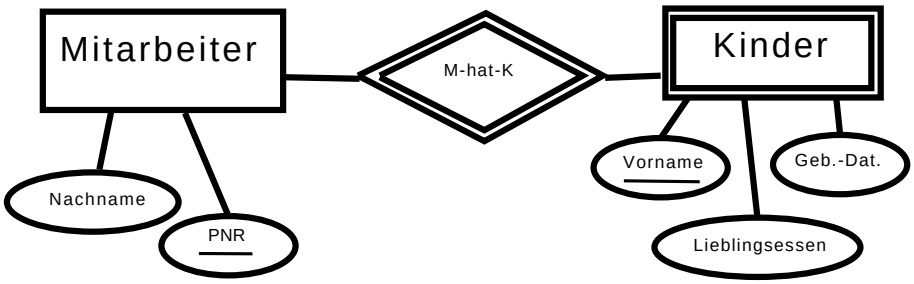


Abbildung 2.1: Schwache Entitätsmengen (unvollständig)

Dieses Beispiel erlaubt eine **zweite Lösung**, die ohne die Benutzung der schwachen Entitätsmengen auskommt: In der Entitätsmenge Mitarbeiter wird ein weiteres, mehrfaches Attribut Kinder hinzugefügt. Dieses Attribut selbst ist zusammengesetzt aus den Bestandteilen Vorname, Geb.-Dat., Lieblingsessen.

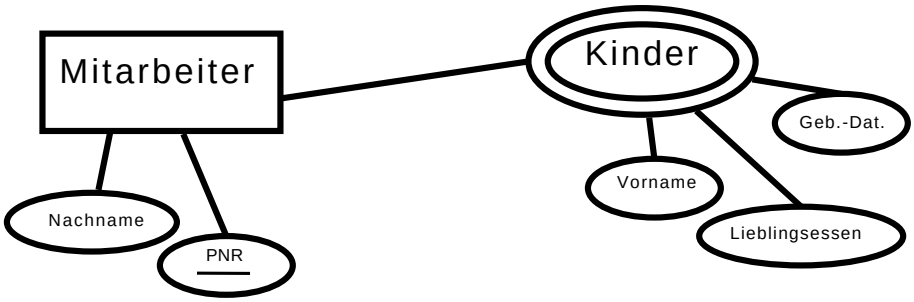


Abbildung 2.2: Betriebskindergarten mit mehrfachem Attribut

Hier entfällt natürlich das Problem der Schlüsselfindung.

Eine genaue Betrachtung beider Lösungen offenbart, dass sie semantisch nicht gleichwertig sind. Die Modellierung als schwache Entitätsmenge beinhaltet die Aussage, dass Kinder etwas Eigenständiges sind. Bei der zweiten Lösung werden sie als Eigenschaft eines Mitarbeiters aufgefasst. Auch wird dabei ein Kind mit genau einem Mitarbeiter in Beziehung gebracht. Der Lösung als schwache Entitätsmenge ist hier flexibler. Sie erlaubt, dass ein Kind mit mehreren Mitarbeitern in Beziehung steht, etwa wenn beide Eltern in der Firma arbeiten. Dies hat Konsequenzen bei einem möglichen Ausscheiden eines Elternteils aus Firma. Bei der Lösung mit schwachen Entitätsmengen ist dies ohne Probleme möglich, bei der Lösung mit dem mehrfachen Attribut ist in diesem Fall eventuell eine Zuordnung der Kinder zu dem anderen Elternteil notwendig. Das Problem der Vielfachheit einer Beziehung wird noch genauer zu untersuchen sein.

2.3.2 Kein Schlüsselkandidat?

Theoretisch dürfte es keine Entitätsmengen geben, zu denen kein Schlüsselkandidat gefunden werden kann (vgl. Abschnitt 2.1.3, S.17), denn in der Miniwelt werden verschiedene Objekte auch unterschieden. Wenn dies im Modell nicht mehr zutrifft, können zwei Gründe dafür genannt werden:
Grund 1: die Einhaltung der Empfehlung, einfache Attribute im Primärschlüssel zu verwenden,
Grund 2: Es wurden wesentliche Merkmale der Objekte nicht als Attribute im Modell berücksichtigt. Dies kann ein Modellierungsfehler genannt werden. Auch wenn dieser Standpunkt, einen Fehler anzunehmen und eine Modellierung mit weiteren Merkmalen der realen Objekte erneut zu versuchen, logisch ist, gibt es doch praktische Gründe, in einem solchen Fall nach einer weniger aufwändigen Lösung zu suchen. Ein solcher Grund könnte sein, dass die Werte der Attribute, die den Schlüsselkandidaten bilden würden, in der praktischen Anwendung nicht zur Verfügung stehen oder dass die Datenerhebung zu aufwendig wäre oder ... oder ... . Das folgende Beispiel diene zur Illustration.

Beispiel:
Eine Entitätsmenge „Studenten“ soll modelliert werden, zur Verfügung stehen die Attribute Name, Vorname, Geburtsdatum, Geburtsort, Anschrift mit Wohnungsnummer. Zieht man solche Millionenstädte wie Peking mit in Betracht, so wird klar, dass alle genannten Attribute möglicherweise für die Bildung eines Schlüsselkandidaten gebraucht werden. Hier finden sich leicht Beispiele, dass Name, Vorname, Geburtsdatum, Geburtsort nicht die Eindeutigkeit der beschriebenen Person gewährleisten, während es nahezu ausgeschlossen ist, dass zwei Personen in allen Attributen übereinstimmen. Damit währe zwar die Bildung eines Schlüsselkandidaten aus allen genannten Attributen möglich, aber praktisch nicht sinnvoll (beispielsweise wegen häufiger Veränderung der Anschrift durch Umzug) Ein solcher zusammengesetzter Schlüssel tritt auch im täglichen Leben auf, ist also überhaupt nicht an die Informatik gebunden. Im universitären Leben wurde deshalb schon vor langer Zeit etwas eingeführt, was diese Attributkombination ersetzt: die Matrikelnummer. Ursprünglich war dies die Nummer eines Eintrags im Matrikelbuch. Auf diesen Eintrag wurde bezug genommen, um einen Studenten zu benennen. Natürlich ist diese Matrikelnummer leichter handhabbar als die o.g. Attributkombination.
Ein anderes Beispiel einer solchen künstlichen Schlüsselkonstruktion gab es in der DDR in Form der Personenkennzahl, wo aus Geburtsdatum, Geschlecht, Geburtskreis ([10]) und einer dreistelligen fortlaufenden Nummer ein künstlicher Schlüsselkandidat für alle Bürger erstellt wurde.
Aber auch die künstlichen Schlüsselkandidaten müssen kritisch hinsichtlich ihrer Gültigkeit hinterfragt werden. Die Matrikelnummer, wie sie heute verwendet wird, ist 6stellig (+ eine Prüfziffer). Dies macht klar, dass sie nicht hochschulübergreifend

[10] Der *Kreis* war eine Verwaltungseinheit.

sein kann und selbst an einer Hochschule nach einigen Jahren Wiederverwendungen auftreten. Soll also ein Student in der Bundesrepublik Deutschland eindeutig festgelegt werden, muss beispielsweise die Kombination *(Hochschule, Immatrikulationsjahr, Matrikelnummer)* verwendet werden. Und im Falle der Personenkennzahl hat nun der Lauf der Geschichte dafür gesorgt, dass der Punkt des Ausschöpfens aller Möglichkeiten nie erreicht wird. Aus dem Beispiel der Studentendaten kann der Schluss gezogen werden, dass durchaus ein künstlich erzeugter Schlüsselkandidat sinnvoll sein kann ([11]), jedoch muss auch warnend erwähnt werden, dass dadurch die Gefahr eines Verlustes an Semantik besteht: Wählt man *(Name, Vorname, Geburtsdatum, Geburtsort, Anschrift)* zum Primärschlüssel, ist jede Kombination der Attributwerte einzig in der Datenbank. Ist die Matrikelnummer der Primärschlüssel, muss die Einzigkeit der Wertekombination der genannten Attribute zur zusätzliche Bedingungen ausdrücklich gefordert werden. Deshalb sollte ein künstlicher Primärschlüssel nur ein letzter Ausweg sein und deshalb (möglichst) nicht im E/R-Modell angewendet werden. Er kann in einem späteren Stadium der Modellierung immer noch in ein Modell eingebracht werden.

2.4 Modellierung weiterer Eigenschaften

Schon bei der Einführung von Rollen wurde deutlich, dass in der Beschreibung einer Miniwelt noch mehr Informationen enthalten sind als mit Entitätsmengen und Relationship-Mengen ausgedrückt wird. Insbesondere sollen Beziehungen genauer beschrieben werden. Das oben genannte Beispiel der Kinder im Betriebskindergarten (Abschnitt 2.14:) gibt Hinweise, dass insbesondere die Vielfachheit einer Beziehung genauer dargestellt werden sollte.
Es existieren mehrere Möglichkeiten der Charakterisierung, die nicht äquivalent sind, also unterschiedliche Merkmale ausdrücken. Es bedarf deshalb einer Vereinbarung, wenn nicht alle Möglichkeiten genutzt werden sollen. Desweiteren darf nicht unerwähnt bleiben, dass hier keine Kompatibilität zum UML-Modell besteht, obwohl bei den Kardinalitätsrestriktionen die Notation zum Verwechseln ähnlich ist.

2.4.1 Grad einer Relationship-Menge

Definition 2.16: *Der* **Grad einer Relationship-Menge** *(ohne eigene Attribute) ist die Anzahl der Entitätsmengen, die in die durch die Relationship-Menge beschriebene Beziehung eingehen.*

Der Grad einer Relationship-Menge ist stets größer als Eins. Sind n Entitätsmengen beteiligt, nennt man die Beziehung *n-är*.

[11] Eine Untermauerung der Motivation ergibt sich noch aus technischen Fragen in Zusammenhang mit dem Übergang ins Relationenmodell (siehe Abschnitt 3.2, S. 45

Definition 2.17: *Eine Relationship-Menge heißt binär g.d.w sie eine Beziehung zwischen genau zwei Entitätsmengen beschreibt.*([12])

Mathematisch kann eine Relationship-Menge als Teilmenge des Kreuzprodukts der an der Beziehung teilnehmenden Entitätsmengen betrachtet werden. In einem Kreuzprodukt darf natürlich keine Komponente fehlen. Dies bedeutet, es zu jedem Eintrag in einer Relationship-Menge die zugehörigen Einträge in den Entitätsmengen per se geben muss. Diese Feststellung wird sich als wichtig erweisen. Wir werden auf sie beim Übergang zu einem weiteren Modell, dem Relationenmodell, bzw. zu einer Datenbank zurückkommen.

2.4.2 Abbildungstypen

Für binäre Relationship-Mengen (und nur für diese !) ist der Begriff des **Abbildungstyps** erklärt. Es wird zwischen den Typen 1:1, 1:N, N:1 und N:M unterschieden.
Der Abbildungstyp wird im Diagramm durch die Zeichen 1, M, N dargastellt:

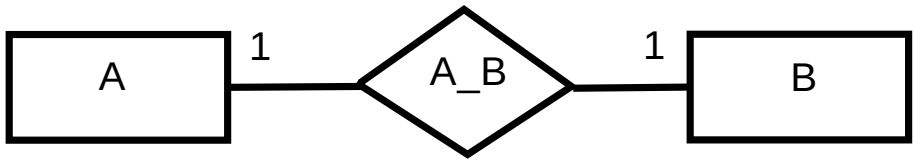


Abbildung 2.3: Abbildung vom Typ 1:1

Definition 2.18: *Die Beziehung* **A** $-$ **B** *ist vom* **Typ 1:1** *g.d.w.*
ein Element der Entitätsmenge **A** *mit höchstens einem Element von* **B und**
ein Element der Entitätsmenge **B** *mit höchstens einem Element von* **A**
ein Paar bildet.

Mit anderen Worten:
Die Abbildung $\varphi : D(\varphi) \mapsto \mathbf{B}$ mit dem Definitionsbereich $D(\varphi) \subseteq \mathbf{A}$ ist eine Funktion und ebenso die Umkehrabbildung $\chi : D(\chi) \mapsto \mathbf{A}$ mit $D(\chi) \subseteq \mathbf{B}$.

Wie in der Definition durch $D(\varphi) \subseteq \mathbf{A}$ ($D(\varphi)$ ist Teilmenge von **A**) ausgedrückt wird, muss nicht jedes Element von **A** an der Beziehung teilnehmen. Dieser Schluss kann auch für **B** gemacht werden. Eine solche optionale Teilnahme gilt auch für die folgenden Abbildungstypen.

[12] Diese müssen nicht verschieden sein.

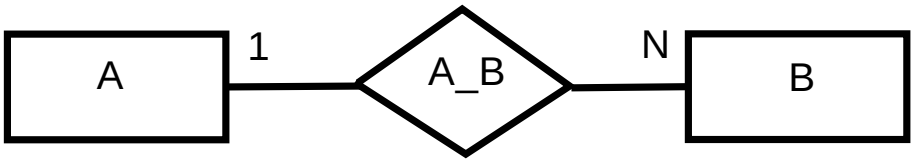


Abbildung 2.4: Abbildung vom Typ 1:N

Definition 2.19: *Die Beziehung* **A** − **B** *ist vom* **Typ 1:N** *g.d.w.*
ein Element der Entitätsmenge **A** *eventuell mit mehreren Elementen von* **B und**
ein Element der Entitätsmenge **B** *mit höchstens einem Element von* **A**
ein Paar bildet.

Hier ist die Abbildung $\mathbf{B} \mapsto \mathbf{A}$ eine Funktion, während die Umkehrabbildung mengenwertig ist.

Der Abbildungstyp N:1 entspricht dem Typ 1:N, wenn die Betrachtungsrichtung umgekehrt wird:

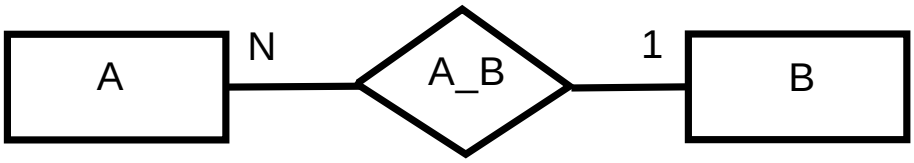


Abbildung 2.5: Abbildung vom Typ N:1

Definition 2.20: *Die Beziehung* **A** − **B** *ist vom* **Typ N:1** *g.d.w.*
ein Element der Entitätsmenge **A** *mit höchstens einem Elementvon* **B und**
ein Element der Entitätsmenge **B** *eventuell mit mehreren Elementen von* **A**
ein Paar bildet.

Der Typ M:N ist der allgemeinste Fall. Hier liegt in jeder Richtung Mehrdeutigkeit bzw. Mengenwertigkeit vor.

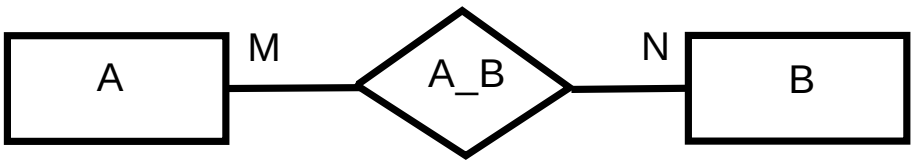


Abbildung 2.6: Abbildung vom Typ M:N

Definition 2.21: *Die Beziehung* **A** − **B** *ist vom* **Typ M:N** *g.d.w.*
ein Element der Entitätsmenge **A** *eventuell mit mehreren Elementen von* **B und**
ein Element der Entitätsmenge **B** *eventuell mit mehreren Elementen von* **A**
ein Paar bildet.

Die Semantik dieser Typen entspricht genau der aus der Mathematik bekannten. Wegen dieses mathematischen Hintergrundes ist der Begriff des Abbildungstyps auf binäre Beziehungen beschränkt.

2.4.3 Kardinalitätsrestriktionen

Mit den Abbildungstypen wird nicht die Frage der Teilnahme an der Beziehung geklärt. Aussagen, wie oft eine konkrete Entität an einer Beziehung teilnimmt, werden durch die Kardinalitätsrestriktionen ausgedrückt. Im Diagramm wird dies mit Hilfe der (min,max)-Notation dargestellt.

Beispiel 2.22: *Gegeben seien zweit Entitätsmengen „Männer“ und „Frauen“ jeweils mit dem Schlüsselattribut „Ausweisnummer“ und eine Beziehung „verheiratet-mit“. Legt man deutsches Eherecht zu grunde, ist der Abbildungstyp 1:1. Dieser Abbildungstyp macht jedoch keine Aussage über die Teilnahme an der Beziehung. Ein Mann (Eine Frau) kann an der Beziehung teilnehmen, muss aber nicht teilnehmen. Allerdings kann er/sie höchstens einmal teilnehmen.*
Dieser Sachverhalt wird im E/R-Diagramm wie folgt ausgedrückt:

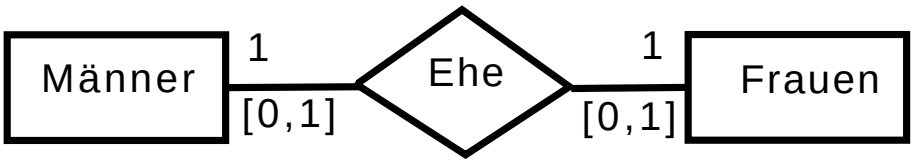
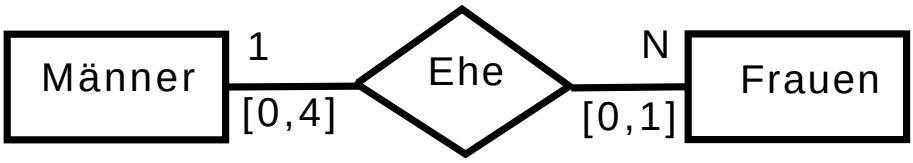


Abbildung 2.7: Kardinalitätsrestriktionen - 1

In an anderem Ort gültigen Rechtsauffassungen kann ein Mann mit bis zu vier Frauen (gleichzeitig) verheiratet sein, die Frau mit höchstens einem Mann. Dies modellierend wird „verheiratet-mit“. sofort eine 1:N -Beziehung. Ein Mann kann jetzt 0 bis 4 Mal in der Relationship-Menge vertreten sein, bei der Frau gilt weiter die Restriktion [0,1].



Kardinalitätsrestriktionen - Vier Ehefrauen möglich

Allgemein gilt: Steht neben der Entitätsmenge **A** an der Beziehung **A** − **B** das Symbol [a,b], so muss jedes Element x aus **A** mindestens a Mal und darf höchstens b Mal an der Beziehung teilnehmen.

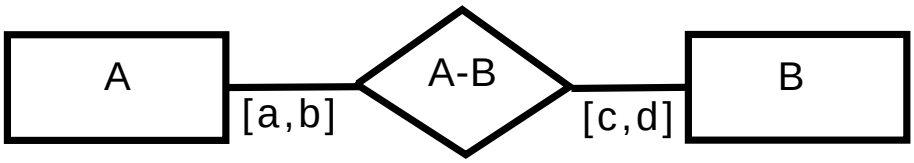


Abbildung 2.8: Kardinalitätsrestriktionen - allgemein

Statt b steht ein Stern, wenn die Anzahl nicht nach oben beschränkt ist (beliebig oft) oder keine konkrete obere Grenze angegeben werden soll.
Eine untere Grenze Null (0) zeigt also eine optionale Teilnahme an, eine 1 (oder jede andere positive Zahl) an dieser Stelle bedeutet die obligatorische Teilnahme, daraus ergibt sich dann beim Datenbankentwurf ein Pflichtfeld.

Mit dieser Notation und den Abbildungstypen kann das oben gegebene Kindergartenbeispiel abgeschlossen werden.

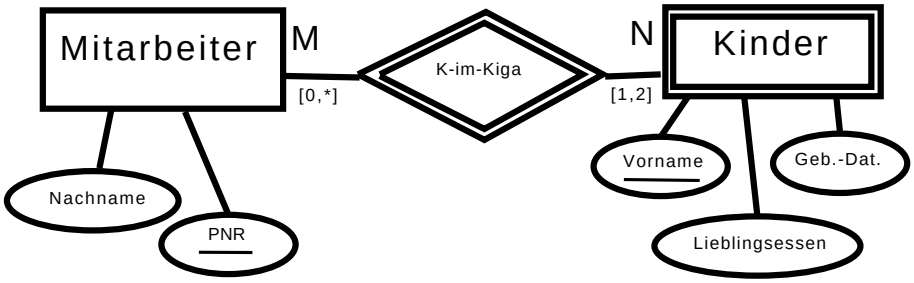


Abbildung 2.9: Betriebskindergarten

Konkret modellieren die Kardinalitätsrestriktionen in dem Beispiel, dass ein Mitarbeiter keine Kinder haben muss und beliebig viele Kinder haben kann, die in den Betriebskindergarten gehen. Aus Sicht eines Kindes wird gefordert, dass mindestens ein Elternteil Mitarbeiter ist, aber auch beide Elternteile können Mitarbeiter sein.
Die (min,max)-Notation wurde 1974 durch Jean-Raymond Abrial eingeführt.

In Verbindung mit Relationship-Mengen vom Typ 1:1, 1:N oder N:1 (siehe Abschnitt 2.4.2, Seite 29 ergeben sich Einschränkungen für die zulässigen Werte in der (min,max)-Notation. . Diese Abbildungstypen haben in mindestens einer Richtung funktionalen Charakter, d.h. zu einem Wert gibt es höchstens ein Bild oder,

anders ausgedrückt: Wenn eine Abbildung einer Entitätsmenge **A** in eine andere **B** eine Funktion ist, darf jedes Element von **A** höchstens einmal an der Abbildung teilnehmen. Deshalb gibt es dann für die (min,max)-Notation auf der **A**-Seite der Relationship-Menge nur die beiden Möglichkeiten (0,1) oder (1,1)

2.4.4 Kardinalitätsrestriktionen bei n-ären Beziehungen

Unglücklicherweise ändert sich bei n-ären Beziehungen die Beschreibung der Semantik der Notation. Wegen Satz 2.13: (Seite 23) ist eine dreistellige Beziehung nicht

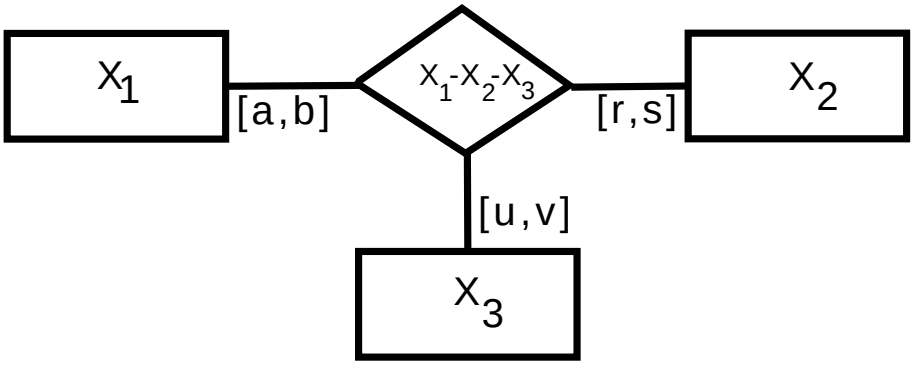


Abbildung 2.10: Dreistellige Beziehung

gleichwertig zu drei binären Beziehungen: aus einem Tripel einer dreistelligen Beziehungen lassen sich stets die drei Paare der zweistelligen Beziehungen ableiten, aber aus drei unabhängigen Paaren muss sich kein Tripel bilden lassen.

Definition 2.23: Kardinalitätsrestriktionen bei n-ären Beziehungen
(Zitat([13]))
Allgemein gilt, dass in einer n-ären Beziehung zwischen dem Entitätstyp $\mathbf{X}_1$ *und* $n-1$ *weiteren Entitätstypen* $\mathbf{X}_2,\ldots,\mathbf{X}_n$ *dem Entitätstyp* $\mathbf{X}_1$ *das Min-Max-Paar* (a,b) *zugeordnet wird, falls jede Entität aus der Entitätsmenge* $\mathbf{X}_1$ *mit mindestens a und höchstens b* $n-1$*-Tupeln aus* $\mathbf{X}_2 \times \cdots \times \mathbf{X}_n$ *in Beziehung steht, d.h. falls es für jedes* $x_1 \in \mathbf{X}_1$ *in der Ausprägung der betreffenden Relation zwischen a und b Tupel gibt, in denen* x_1 *(an erster Stelle) vorkommt - also Tupel der Form* $(x_1,\ldots)$*.*

Erst bei der allgemeinen Definition für n-äre Relationen tritt der wesentliche Charakter der Min-Max-Notation, Ausprägungen der Relation zu zählen, deutlich zu Tage. (Zitatende.)

[13] Quelle: Wikipedia, Stichwort: Min-Max-Notation.

2.5 Defizite im E/R-Modell

- Das E/R-Modell unterstützt die Konzepte der objektorientierten Modellierung nicht. Folglich fehlen die Möglichkeiten, Vererbungen oder Komponentenbeziehungen direkt zu unterstützen. Das Modell kann jedoch dahingehend erweitert werden, indem spezielle Relationen für diesen Zweck erklärt werden.
- Relationship-Mengen können nur zwischen Entitätsmengen existieren, das macht es unter anderem unmöglich, Bedingung der folgenden Art zu beschreiben: Relationship-Menge R ist Teilmenge der Relationship-Menge S, wobei R und S jeweils zwischen den Entitätsmengen A und B erklärt sind. Zur Ergänzung des Modells können *komplexe Entitätsmengen* eingeführt werde.

In all diesen Fällen kann die Symbolik aus dem UML-Modell sinnfällig als Erweiterung des E/R-Modells benutzt werden (Vererbung, Komponenten, Kommentare). Dies wird im Kapitel Ergänzungen (Seite 35) kurz vorgestellt werden.

- Das Modell ist nicht formal spezifiziert, so mit ergeben sich große Probleme bei der Weiterverarbeitung in ein einem Computer. Diese Probleme zeigen sich einmal beim Entwurfsprozess in der mangelhaften Unterstützung durch Werkzeuge, was insbesondere bei großen Entwürfen mit vielen Entitätsmengen und Relationship-Mengen allein schon die Anordnung der Elemente zu einem Problem werden lässt. Die o.g. Forderung nach Beschreibung von Kontext und Semantik wird, da nicht explizit in der graphischen Notation vorgeschrieben, meist vernachlässigt und dies ist dann eine Ursache der Probleme bei Datenintegrationen.

2.6 Bewertung

Trotz der beschriebenen Defizite stellt des E/R-Modell ein wichtiges Hilfsmittel des Datenbankentwurf dar. Ein wichtiger Vorteil ist die leichte Erlernbarkeit. Da es leicht nach einem einfachen Regelsystem in einen relationalen Datenbankentwurf transformiert werden kann (s. Seite 45), welcher dann seinerseits quasi eins-zu-eins in SQL-Anweisungen zur Implementierung einer Datenbank umgesetzt werden kann, ist der Datenbankentwurf nach dem E/R-Modell dann zu empfehlen, wenn eine relationale Datenbank erzeugt werden soll.
Da insbesondere die Attributbildung kaum eingeschränkt ist, können Attribute konstruiert werden, die sehr gut an die Anforderungen der Anwendungen angepasst sind ([14]).

Leider ist die Unterstützung durch Entwicklungswerkzeuge schlecht. Teilweise sind diese auch schon zu sehr auf eine Umsetzung des Entwurfs als

[14] Es gibt noch keine Einschränkungen durch Datentypen.

SQL-Anweisungen ausgerichtet. Dies schränkt die Mächtigkeit der Attributkonstruktionen ein.
Der Nachteil der mangelnden Toolunterstützung ist jedoch nur die Folge der tiefer liegenden Defizite wie der fehlenden formalen Beschreibung. Fairerweise muss jedoch klargestellt werden, dass zur Zeit der Entstehung des Modells vor ca. 30 Jahren diese nicht Hauptaugenmerk der Entwicklung war. Die Arbeitsweise bei der Erstellung von Software u.ä. war so, dass zunächst ein Entwurf in Papierform entstand, z.B. durch E/R-Entwürfen oder bei Programmen durch Nassi-Shneiderman-Diagramme ([15]). Solche visuell unterstützten Entwurfstechniken können sehr effizient sein.
Viel schwerer wiegt die Tatsache, dass das Modell nicht formal beschrieben ist, unter den Anforderungen des Jahres 2008. Ein wichtiger Gegenstand der Datenbankforschung ist die Datenintegration. Unter den Anforderungen dieses Themas wäre eine formale Beschreibung des Modells, der Attribute usw. in einer maschinenlesbaren Form ein wichtiger Aspekt und eine Voraussetzung für automatisierte Schritte der Weiterverarbeitung bzw. einer automatisierten Integration relationaler Datenbanken.

Das E/R-Modell ist, wie schon das Jahr seiner Erstpublikation vermuten lässt, weit vor der Zeit des objektorientierten Paradigmas entstand. Wie oben bemerkt wurde , fehlt die explizite Unterstützung dieses Paradigma. Die symbolische Darstellung der in der Objektorientierung viel verwendeten semantischen Beziehungstypen *is-a* mit der Bedeutung einer Teilmengenbeziehung und *part-of* bzw. *component-of*, mit eine Entität zum Bestandteil einer anderen erklärt wird, können jedoch als Erweiterung in das Modell aufgenommen werden.
Einige Lösungsansätze werden im folgenden Abschnitt kurz vorgestellt.

2.7 Ergänzungen / Erweiterungen

Einige der Defizite des E/R-Modells lassen sich kompensieren, indem Elemente aus UML eingebracht werden. So kann eine Beziehung mit der Bedeutung „is-a“ (Vererbung) statt mit dem Relationship-Symbol mit dem Symbol aus UML für diese Semantik ($(-\triangleright)$) dargestellt werden. Der Abbildungstyp ist dann 1:1, die Kardinalitätsrestriktion an der Untermenge ist stets $[1,1]$, auf der Seite der Obermenge kommen nur $[0,1]$ (die Untermenge A ist eventuell eine echte Teilmenge von B - s. Abb. 2.11) oder $[1,1]$ (jedes Element der Obermenge A gehört auch zur Untermenge B) in Frage.

Festlegung:
Für Beziehungen mit der is-a-Semantik wird dabei noch zusätzlich vereinbart,

[15] dazu existiert Toolunterstützung

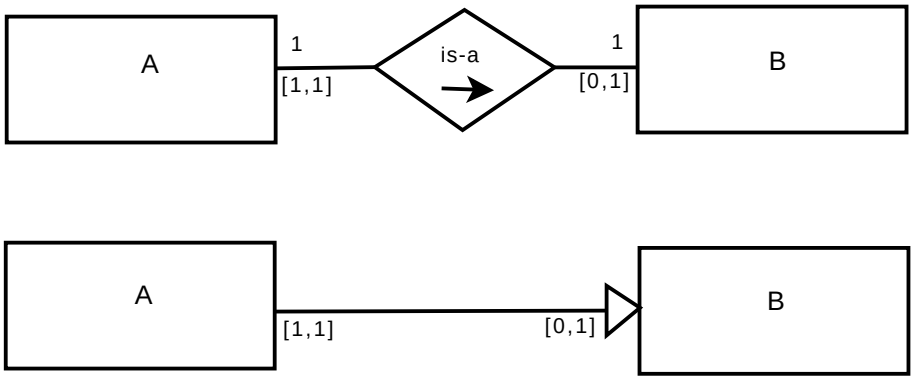


Abbildung 2.11: Beispiel für UML-Symbole

- dass alle Attribute der Obermenge auch Attribute der Untermenge sind und
- dass alle Beziehungen, die die die Obermenge eigeht auch für die Untermenge bestehen.

■

Die soll stets gelten, auch wenn oder obwohl die Attribute und Beziehungen nicht explizit als Attribute und Beziehungen der Untermenge angegeben sind. Dies entspricht der Bedeutung des Begriffs „Vererbung“, oder auch der mehr mathematischen Charakterisierung als Teilmengenbeziehung: Jedes Element einer Untermenge gehört auch zur Obermenge, hat also deren Merkmale. Daraus folgt auch, dass die Bildung des Primärschlüssels für die der Untermenge entsprechenden Entitätsmenge kein Problem ist: Die Attribute, die in der Obermenge den Primärschlüssel bilden, erfüllen die Bedingungen für einen Schlüsselkandidaten natürlich auch auf einer Teilmenge der Entitäten. Die Entitätsmenge „Professoren“

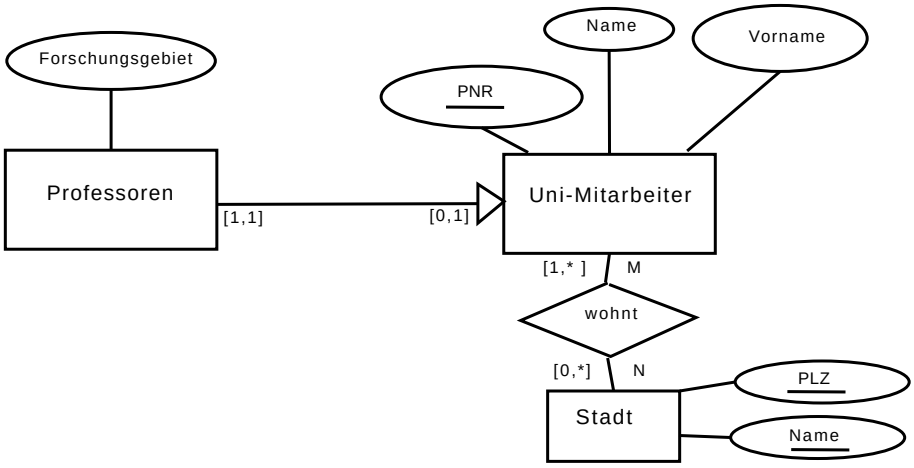


Abbildung 2.12: Attribute und Beziehungen bei Vererbung

in Abbildung 2.12 (Seite 36) hat also auch die Attribute *Personal-Nr, Name* und *Vor-*

name und jedes seiner Elemente muss an der Beziehung *wohnt* mindestens einmal teilnehmen. Da die Personalnummer *PNR* Primärschlüssel für alle Uni-Mitarbeiter ist, ist kann sie diese Funktion auch für die Teilmenge der Professoren erfüllen.

Durch Übernahme der entsprechenden Symbole für die Komponentenbeziehungen lässt sich weitere Semantik hinzufügen. Diese Erweiterungen erhöhen zweifelsohne die visuelle Aussagekraft der Diagramme. Es darf jedoch nicht verschwiegen werden, dass damit einige Schwierigkeiten nur auf spätere Entwicklungsschritte verschoben werden. Weder Vererbung noch Komponentenbeziehungen vom Relationenmodell direkt mit dieser Semantik unterstützt werden. Das bedeutet eine etwas komplexere Umsetzung dieser Beziehungen (siehe Abschnitt 3.2.2 über Partitionierung).

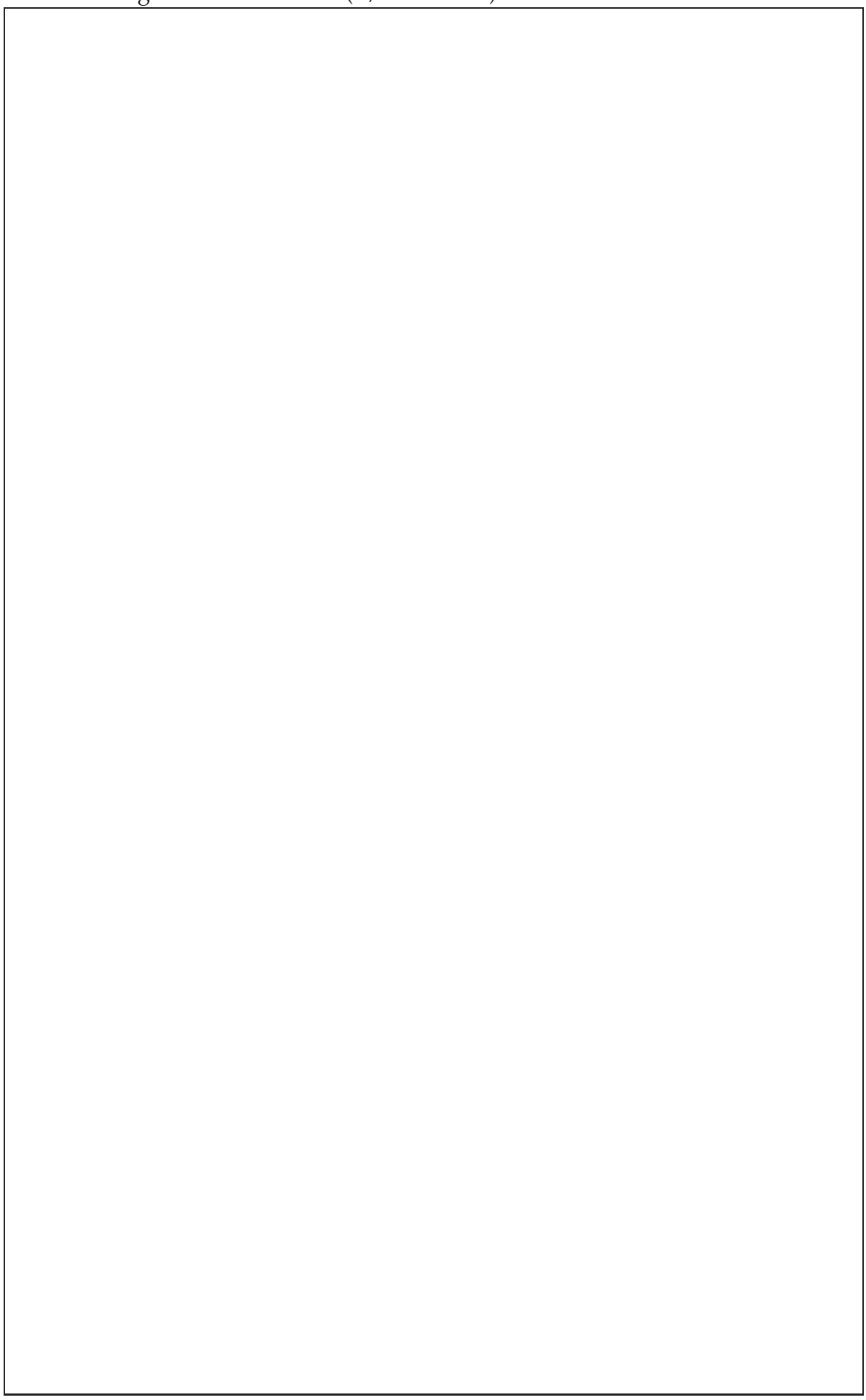
2.8 Beispiel für einen E/R-Entwurf

Mit dem folgenden Beispiel besteht die Gelegenheit, die Komponenten des E/R-Modells praktisch einzusetzen. Die Lösung befindet sich am Ende des Lesehefts im Anhang (Seite 63).

Aufgabe 2.24: *Vorgelegt sei folgende Banken-Miniwelt:*
Es gibt Banken und Kunden.
Eine Bank, welche durch ihre Bankleitzahl und ihren Namen beschrieben wird, besitzt mindestens eine Zweigstelle. Jede Zweigstelle hat eine Adresse und eine innerhalb der Bank eindeutige Zweigstellennummer. Eine Zweigstelle verwaltet Konten. Wir unterscheiden Guthaben- und Kreditkonten. Zu beiden Arten gibt es wiederum Unterarten (z. B. bei den Guthabenkonten Girokonten und Sparbücher, bei Kreditkonten z.B. Konten für allgemeine Kredite und Konten für Baukredite).
Jedes Konto hat eine innerhalb der Miniwelt eindeutige Kontonummer, einen Kontostand und einen Zinssatz. Bei Kreditkonten gibt es eine vereinbarte monatliche Tilgungssumme. Zur Sicherheit für die Bank müssen für jedes Kreditkonto mindestens zwei Kontoinhaber angegeben werden.
Von den Kunden werden Vorname, Nachname, Geburtsdatum, Wohnanschrift und eine bankintern eindeutige Personenkennzahl gespeichert. Kunde wird, wer mindestens ein Konto hat. Ein Kunde kann mehrere Guthaben- und / oder Kreditkonten haben. Es darf keine Kunden geben, die nur ein Kreditkonto haben oder mehr als drei Kreditkonten haben.
Aufgabe: *Ist ist ein E/R-Diagramm zu erstellen, welches die Vorgaben modelliert. Dabei sollen die aus UML stammenden Erweiterungen nicht benutzt werden.*

Für die Lösung ist die folgende Seite reserviert, eine kommentierte Musterlösung wird im Anhang (Seite 63) angegeben. Natürlich besteht die Empfehlung, vor dem Lesen der Musterlösung eine eigene Lösung zu erarbeiten.

Meine Lösung: Banken-Miniwelt (E/R-Entwurf)



3 Das Relationenmodell

In der Arbeit [2] aus dem Jahre 1970 hat *Ted Codd* , der für IBM Research tätig war, das Relationenmodell vorgestellt. Es enthielt ein konzeptionell neues Herangehen an die Datenbankmodellierung. Vor dem Relationenmodell gab es schon mehrere Versuche, Modelle für die Datenbankarbeit zu entwickeln. So entstanden in den sechziger Jahren des vergangenen Jahrhunderts das *hierarchische Modell* und das *Netzwerkmodell* . Sie wurden in den beiden folgenden Jahrzehnten in Datenbankverwaltungssystemen (DBVS, *engl.* DBMS) implementiert. Nachteilig in beiden Modellen waren die sehr aufwendigen und tiefgreifenden Änderungen in der Datenbank, die notwendig wurden, wenn kleine Veränderungen in der Datenstruktur zu realisieren waren.
Das von *Codd* angeregte Modell zeichnet sich durch Klarheit und Einfachheit aus. *Codd* benutzte die Relation als Datenstruktur, die leicht an verschiedene Anforderungen angepasst werden kann. Hinzu kommt, dass mathematische Grundlagen für die Arbeit mit dem Modell angegeben werden können, die Relationenalgebra. Diese Algebra bildet den Gegenstand eines folgenden Bandes des Lese-und Übungsbuchs.
Das Relationenmodell und damit auch alle seine Implementierungen sind mengenorientiert. In Mengen ist keine Ordnung der Elemente definiert, deshalb ist in den auf dem Relationenmodell aufbauenden Datenbankverwaltungssystemen eine deskriptive Anfragebeschreibung angemessen. Sie beschreibt nur, was in der Anfrage gefunden werden soll, und nicht, wie dies zu geschehen hat. Somit kann ein Datenbankverwaltungssystem mehrere Ausführungspläne zu einer Anfrage erstellen, diese bewerten und einen günstigen Plan zur Ausführung wählen. Die relationale Algebra liefert einen Teil des theoretischen Fundaments für die Erstellung solcher Pläne.
Das relationale Modell wurde seither in vielen kommerziellen Produkten umgesetzt, diese Implementierungen arbeiten heute sehr zuverlässig und performant. Sie sind als relationale Datenbankverwaltungssysteme weitverbreitet.

Beim Relationenmodell ist im Vergleich zum E/R-Modell die Zahl der Grundkonstrukte reduziert: es gibt nur Relationenmengen. Auch die Varianten der Attributbildungen wurden reduziert. Deshalb wird zu diskutieren sein, wie ein Entwurf nach dem E/R-Modell verlustfrei in das Relationenmodell transformiert werden kann. Aber auch die Transformation aus dem Relationenmodell in ein E/R-Modell, die zwar in der Praxis kaum benötigt wird, soll kurz diskutiert werden. Diese Diskussion ermöglicht nochmals einen Blick darauf, wie einige Eigenschaften der Miniwelt aus dem E/R-Modell in das Relationenmodell gebracht wurden. Diese Er-

kenntnisse können beispielsweise bei der Datenintegration ausgenutzt werde.
Die Einfachheit des Modells bietet andererseits die Möglichkeit, die semantische Korrektheit des Modells mit Hilfe der Normalformenlehre einer Überprüfung zu unterziehen. Da es schon im E/R-Modell keine formale Beschreibung der Semantik gibt und sich dieser Zustand in Relationenmodell nicht ändert, kann auch diese Überprüfung nur informell erfolgen, also durch das Verstehen durch einen Menschen.

3.1 Modellbeschreibung

3.1.1 Attribute

Ausgehend von der Attributdefinition / Attributbeschreibung des E/R-Modells (siehe Seite 13) ergibt sich im Relationenmodell folgende Einschränkung: alle Attribute sind einfach und zwar im doppelten Sinn des Wortes:

1. Es gibt keine mehrfachen Attribute.
2. Es gibt keine zusammengesetzten Attribute.

Definition 3.1: *Attribute mit den eben genannten Eigenschaften heißen* **atomar** *oder* **einfach** *([1])*

Beim Übergang vom E/R-Modell zum relationalen Modell müssen also ggf. einige Attributkonstruktionen des E/R-Modells aufgelöst werden.

3.1.2 Relationenmengen

Grundlegende Merkmale - Begriffe

Definition 3.2: *Gegeben seien eine Menge $\mathcal{A}$ von Attributen $\mathcal{A} = \{A_i\}_{i=1}^{k}$ für eine geeignete natürliche Zahl k. Eine* **Relation** *über $\mathcal{A}$ ist eine Teilmenge des Kreuzprodukts der A_i, also eine Menge von Tubeln $(a_1, a_2, a_3, ..., a_k)$ mit $a_i \in \mathcal{W}(A_i), i = 1, ..., k$, wobei $\mathcal{W}(A_i)$ den Wertebereich (Wertevorrat) von A_i bezeichnet.*

Während in der Menge $\mathcal{A}$ keine Ordnung bzw. Reihenfolge existiert, ergibt die Kreuzproduktbildung ein (geordnetes) Tupel. Um also das Ergebnis der Kreuzproduktbildung semantisch beschreiben zu können, muß die Reihenfolge in der die Elemente von $\mathcal{A}$ in dem Kreuzprodukt auftreten, mit festgehalten werden. Dies führt zu folgender Definition:

[1] in der Sprache der in einem späteren Band zu besprechenden Normalformenlehre sind dies genau die Eigenschaften der ersten Normalform, damit kann auch formuliert werden: Ein Entwurf nach dem Relationenmodell ist in erster Normalform.

Definition 3.3: Relationsschema
Ein Relationsschema $\mathbf{R}(A_1, A_2, A_3, ..., A_n)$ *besteht aus einem* **Relationsnamen R** *und einer* **Liste** *von paarweise verschiedenen Attributnamen* $(A_1, A_2, A_3, ..., A_n)$*, wobei die zu den Namen gehörigen Attribute definiert sein müssen.*
Zur vollständigen Beschreibung eines Relationsschemas wird noch (analog zum Vorgehen bei Entitätsmengen die Beschreibung der Semantik und des Kontextes benötigt.

Die Anzahl n der Attributnamen heißt **Grad** *des Relationsschemas.*

Das Relationsschema beinhaltet alle strukturellen Metainformationen einer Relation.
$\mathbf{R}(A_1, A_2, A_3, ..., A_n)$ dient auch zur Kurzbeschreibung eines Relationsschemas, wenn keine Missverständnisse auftreten, wird abkürzend nur der Name **R** angegeben.

Definition 3.4: *Die durch die Namen der Liste* $(A_1, A_2, A_3, ..., A_n)$ *bezeichneten Attribute heißen* **Attribute des Relationsschemas R**.

Diese Attribute sind auch Attribute im Sinne des Attributbegriffs auf Seite 13 und damit die Liste nicht nur eine formale Aufzählung von Namen darstellt, müssen sie vor ihrer Benutzung erklärt werden (s. ebenda).

Beispiel 3.5: *Die Professoren einer Universität sollen in einer Tabelle erfaßt werden. Die Tabelle soll als Information den Namen, die Personal-Nummer, das Arbeitsgebiet und die Fakulät, an der er tätig ist, umfassen.*

Dies wird durch das folgende Relationsschema beschrieben:
Professoren$(Personal - Nr, Name, Arbeitsgebiet, Fakultät)$ ■

In dem Beispiel werden die Attribute nur durch ihren Namen spezifiziert, die Namen beschreiben hier die Semantik grob, der Wertevorrat wurde nicht angegeben. Ein solches vereinfachtes Vorgehen kann für einen groben Entwurf akzeptiert werden, das Verstehen dieses Schemas ist hier nur gegeben, wenn der Leser ausreichende Kenntnis der Miniwelt hat. Dies bedeutet, das er weiß, dass jeder Professor ein spezielles Arbeitsgebiet hat, auf dem er vorzugsweise forscht und dass jeder Professor zu genau einer Fakultät gehört. Die Fakultäten sind die wichtigsten wissenschaftlichen Einrichtungen einer Universität. Wenn also Attribute der Relation **Professoren** ohne weitere Erklärung benutzt werden sollen, muss die Miniwelt (hier: Aufbau und Arbeitsweise einer Universität) also zum Allgemeinwissen zählen.
Bei einer Implementierung oder bei der komplexer Sachverhalte sind an dieser Stelle Konkretisierungen nötig, die zum Beispiel die verwendeten Wertevorräte

betreffen([2])

Notation

Wie schon in dem vorausgehenden Abschnitt (Beispiel 3.5:) zu sehen ist, wird eine Notation für Ralationenschemata benötigt. Diese soll einige, zum Teil zur schwer gleichzeitig zur realisierende Eigenschaften haben. Zum einen soll sie intuitiv verwendbar sein, zum anderen auch automatisch weiterverarbeitbar sein. Die zweite Eigenschaft kann mit einer formal definierten Sprache gesichert werden. Und letztendlich leistet die in SQL enthaltene DDL ([3]) mit ihren CREATE-XXX-Befehlen genau das gewünschte:

Satz 3.6: SQL als formal definierte Sprache
Die Struktur von Relationsschemata kann mit SQL maschinenverarbeitbar beschrieben werden.

Eine Beschreibung der Semantik ist in SQL nicht vorgesehen.

Sollen Menschen mit den Realationenschemata umgehen, ist die Angabe von SQL-Befehlen wenig anschaulich. Hier sind Vereinfachungen sowie Anleihen bei der Mathematik und im täglichen Leben ein Ausgangspunkt für eine Darstellung. Die Struktur einer Tabelle mit nicht unterteilten Spalten kann beschrieben werden durch einen Tabellennamen und die Angabe des Tabellenkopfes, d.h. durch Auflistung des Spaltennamens. Hinzukommen noch einige weitere Eigenschaften, die in den folgenden Abschnitten bereitgestellt werden, danach wird die Beschreibung ergänzt. Bis dahin (siehe Abschnitt 4.2, Seite 65) wird eine vorläufige (unvollständige) Notation benutzt:

Satz 3.7: Vorläufige semiformale Notation:
Ein Relationsschema mit dem Namen **R** *und Attributen mit den Namen* $A_1, A_2, ..., A_n$ *wird durch*

$$\mathbf{R}(A_1, A_2, ..., A_n)$$

dargestellt.

■

Diese Notation wird noch durch weitere Informationen angereichert. Ein Vorschlag für eine vereinfachte Notation der Möglichkeiten wird im Anhang (Seite 65) gegeben.

[2] Um mit der Relationenalgebra zu arbeiten, sind solche vereinfachten Darstellungen häufig durchaus akzeptabel und ausreichend.
[3] data definition language

Weitere Merkmale

Wenn ein Relationsschema wie in *Definition 3.3:* durch die Liste seiner Attribute definiert ist, so beinhaltet der Begriff „Liste“ eigentlich, dass die Reihenfolge der Attribute wesentlich ist. **Im Relationenmodell trägt die Reihenfolge der Attribute** jedoch **keine Information**, deshalb sind auch alternative, mengenorientierte Definitionen anstelle von *Definition 3.3:* möglich, in denen von einer *Menge von Attributen* gesprochen wird. Die hier gegebene Definition ist praktisch leichter handhabbar, sie widerspiegelt die Tatsache, dass auch die Spalten einer Tabelle in einer Reihenfolge angegeben werden (müssen). Um die Unabhängigkeit von der Reihenfolge der Attribute zu modellieren, wird der Begriff „äquivalente Relationsschemata“ definiert. Mit diesem Begriff soll eine Klasseneinteilung aller Relationsschemata so bewirkt werden, dass in einer Äquivalenzklasse genau die Schemata liegen, die sich nur durch eine unterschiedliche Reihenfolge der Attribute unterscheiden.

Definition 3.8: Äquivalenz von Relationsschemata
Zwei **Relationsschemata R** *und* **S heißen äquivalent**, *wenn sie in den Attributen, als Menge von Attributen betrachtet, übereinstimmen und die Anordnung der Attribute in* **S** *durch eine Permutation der Reihenfolge der Attribute in* **R** *entsteht.*

Diese Definition bedarf des Nachweises ihrer Korrektheit dahingehend, dass durch sie tatsächlich eine Äquivalenzrelation definiert wird:
Offensichtlich ist die Reihenfolge der Attribute in **S** genau dann eine Permutation der Anordnung in **R**, wenn umgekehrt die Reihenfolge in **R** eine Permutation der Reihenfolge in **S** ist (Symmetrie).
Weiter ist jede Relation zu sich selbst äquivalent (Reflexivität).
Und wenn **R** äquivalent **S** und **S** äquivalent zu einer weiteren Relation **T** ist, ist auch **R** äquivalent **T** (Transitivität), da dies für Permutationen allgemein gilt.
Wegen dieser Eigenschaften wird durch Definition 3.8: tatsächlich eine Äquivalenzrelation in der Menge der Relationsschemata definiert. ■

Definition 3.9: Relation (zu einem Relationsschema)
Eine Relation $\mathbf{r}(\mathbf{R})$, *abkürzend* $\mathbf{r}$, *zu einem Relationsschema* $\mathbf{R}(A_1, A_2, ..., A_n)$ *ist eine Menge von Tupeln* $t_i, i = 1, ..., k$*:*

$$\mathbf{r} = \{t_1, t_2, t_3, ..., t_k\} \text{ mit}$$
$$t_i = (w_{i,1}, w_{i,2}, w_{i,3}, ..., w_{i,n})$$
$$\in dom(A_1) \times dom(A_2) \times dom(A_3) \times ... \times dom(A_n). \quad (4)$$

Der **Grad der Relation** *ist gleich dem Grad des zugehörigen Relationsschemas (also n). Die Anzahl k der Tupel heißt* **Kardinalität der Relation**.

Es ist üblich, dass **R** auch als *Name* einer zum Relationsschema $\mathbf{R}(...)$ entstandenen Relation dient. Für den Begriff „Relation“ sind als alternative Bezeichnungen

[4] $\mathcal{X} \times \mathcal{Y}$ bezeichnet das Kreuzprodukt der Mengen $\mathcal{X}$ und $\mathcal{Y}$,

Relationsinstanz oder *Relationszustand* gebräuchlich. Jedes Tupel einer Relation beschreibt eine konkrete Entität oder eine konkrete Beziehung, die in der modellierten Miniwelt existiert ([5]).
Im folgenden wird eine Instanz der Relation **Professoren** zu dem gleichnamigen Schema aus Beispiel 3.5: gegeben:

Professoren

PNR	Name	Arbeitsgebiet	Fakultät
2364	Rahm	Datenbanken	Math.u.Informatik
0342	Beyer	Analysis	Math.u.Informatik
9645	Wartenberg	Kirchengeschichte	Theologie
4714	Cain	Klass. Archäologie	Geschichte

Jede Spalte der Tabelle besitzt einen Eintrag im Tabellenkopf, den Spaltennamen, dieser korrespondiert zu dem entsprechenden Attributnamen des Relationsschemas. Vielfach werden die Namen so gewählt, dass aus dem Namen die Semantik erschlossen werden kann, aber dies ist nicht zwingend vorgeschrieben: Im Beispiel hat das Attribut mit dem Namen *PNR* die Bedeutung Personalnummer. Da die Attribute als atomar vorausgesetzt wurden, werden dem entsprechend nur Tabellen betrachtet, deren Spalten nicht aus Subspalten zusammengesetzt sind.
Jedes Tupel einer Relation entspricht genau einer Zeile in der Tabelle. Eine Relation **r** wurde als Menge von Tupeln definiert, folglich trägt die **Reihenfolge der Tupel** im Modell keine Information, m.a.W. **die Reihenfolge der Tupel einer Relation ist ohne Bedeutung**. In der anschaulichen Interpretation als Tabelle trägt also die Reihenfolge der Zeilen dieser Tabelle keine Bedeutung. Sie ist als zufällig anzusehen.

Um die Komponenten eines Tupels t anzusprechen, wird im Rahmen des Relationenmodells der Name des entsprechenden Attributs in folgender Form verwendet: $t[A_i]$ und $t.A_i$ beschreiben den Wert des Attributs A_i im Tupel t ([6]). Diese Notation ist vergleichbar mit der *call-by-name*-Parameterübergabe bei Programmiersprachen.

3.1.3 Schlüsselkandidaten, Primärschlüssel

Schlüsselkandidaten und Primärschlüssel sind wie im E/R-Modell definiert. Auch der Umgang mit Werten, die eine Unbestimmtheit ausdrücken hat sich nicht

[5] Es ist an dieser Stelle notwendig, einige Anmerkungen zum Begriff **Relation** zu machen. Zunächst wird er in der Mathematik benutzt, hier beschreibt er eine Teilmenge einer Kreuzproduktmenge, also eine Tupelmenge. In Zusammenhang mit Datenbanken wird der Begriff Relation aber auch als Synonym zu Relationsschema benutzt.

[6] wenn $\mathcal{B} = (A_{l_1}, A_{l_2}, ..., A_{l_j})$ eine Teilmenge aller Attribute eines Relationsschema ist, dann sei $t[\mathcal{B}] = (t.A_{l_1}, t.A_{l_2}, ..., t.A_{l_j})$.

geändet. Zu den wesentlichen Eigenschaften des Relationenmodells gehört, dass zu jedem Relationsschema ein Primärschlüssel angegeben werden muss.

Satz 3.10: *In jedem Relationsschema existiert (mindestens) ein Schlüsselkandidat.*

Beweis:
Jede Relation ist nach Definition eine Teilmenge einer Menge (nämlich einer Kreuzproduktmenge), enthält also nach Konstruktion keine Duplikate. Wenn also die Semantik der Attribute keinen Schlüsselkandidaten erlaubt, der weniger als alle Attribute enthält, dann muss die Menge aller Attribute Schlüsselkandidat sein.
■

Ergänzung zur Notation: Primärschlüssel
Für die Darstellung der Primärschlüssel haben sich zwei Alternativen durchgesetzt:

1. Die den Primärschlüssel bildenden Attribute werden durch Vertauschung der Reihenfolge an aufeinanderfolgende Positionen (meist an den Anfang) der Attributliste gebracht und durch eine *gemeinsame Unterstreichung mit einer durchgehenden Linie* gekennzeichnet.
2. Die Schlüsselattribute erhalten den Zusatz **:PS**, der direkt dem Attributnamen folgt. Diese Darstellung bietet sich insbesondere bei einem nicht zusammengesetzten Primärschlüssel an:
   Studenten$(Matrikelnummer : PS, Name, Vorname, Geburtsdatum)$
   an.
   Zusammengesetzte Schlüssel sollen nach der Attributliste unter Benutzung von Klammern angegeben werden.
   $\mathbf{R}(A,B,C,D),(A,B) : PS;$
   mit der Semantik: die Relation **R** hat die Attribute A,B,C,D, davon bilden A und B gemeinsam den Primärschlüssel.

3.2 E/R - Transformation

Bisher wurde nur ein Relationsschema betrachtet. Um praktisch relevante Fragen zu bearbeiten, müssen jedoch mehrere Schemata gleichzeitig untersucht werden. Wie die folgende Überlegung zeigt, gibt es eventuell Beziehungen zwischen diesen Schemata.
Formal erscheint das Relationenmodell weniger ausdrucksmächtig als das E/R-Modell zu sein, da insbesondere mengenwertige und zusammengesetzte Attribute nicht existieren. Im folgenden wird beschrieben, wie durch eine gute (d.h. Semantik erhaltende) Transformation diese Eigenschaften der Attribute auch im Relationenmodell erhalten werden.

Als Verlust beim Übergang vom E/R- zum Relationenmodell bleibt nur das Wegfallen des Unterschieds von Entitätsmengen und Relationenmengen. Eine genauere Betrachtung zeigt jedoch, dass es in der Beschreibung eines Relationsschemas, welches aus einer Relationshipbeschreibung entstanden ist, strukturelle Merkmale gibt, die typisch für diese Herkunft sind, m.a.W. es gibt in einem Relationsschema Merkmale, die auf seine Herkunft aus einer Relationship-Menge hindeuten. Diese Merkmale können ausgewertet werden, wenn aus Relationsschemata wieder ein semantisch gleichwertiger E/R-Entwurf konstruiert werden soll - vgl. Abschnitt 3.4, Seite 59. Sie können auch bei der Strukturerkennung bei Aufgaben der Datenintegration genutzt werden.
Es ist als eine **Standardaufgabe der Datenbankentwicklung** anzusehen, **aus einem E/R-Entwurf einen semantisch äquivalenten Entwurf nach dem Relationenmodell herzuleiten.** Dabei wird - meist unausgesprochen - die **Nebenbedingung** gestellt, **die Anzahl der entstehenden Relationsschemata minimal zu halten.**
Die Transformation eines E/R-Modells in ein Relationenmodell wird einen Satz von Regeln beschrieben, die im folgenden vorgestellt werden. Die Idee ist dabei für jedes Symbol aus dem E/R-Modell eine Vorschrift für die Umformung und die eventuell notwendig werdende Neudefinition von Primärschlüsseln anzugeben. Dabei wird sich zeigen, dass ein Teil der Aufgaben formal bearbeitet werden kann und bei anderen Aufgaben nochmals die Semantik von Relationen und Attributen (in der Miniwelt) hinterfragt werden muss, um die Umformung korrekt vorzunehmen.

3.2.1 Fremdschlüssel

Zum Verstehen dieser Transformationen ist der Begriff des Fremdschlüssels nötig.

Definition 3.11: Fremdschlüssel
Eine Menge $\mathcal{X}$ von Attributen eines Relationsschemas $\mathcal{R}$ ist ein Fremdschlüssel auf ein Relationsschema $\mathcal{S}$ wenn es

1. *eine eineindeutige (nicht nur hinsichtlich der formalen Merkmale, sondern auch mit Bezug zur Semantik) Zuordnung der Elemente von $\mathcal{X}$ zum Primärschlüssel ([7]) von $\mathcal{S}$ gibt und*
2. *es zu jeder in $\mathcal{R}$ auftretenden Wertebelegung der Attribute von $\mathcal{X}$, die keine Unbestimmtheit ausdrückt, eine Tupel in $\mathcal{S}$ mit genau diesen Werten für die Attribute des Primärschlüssels gibt.*

$\mathcal{R}$ heißt **referenzierende Relation** *, $\mathcal{S}$ ist die* **referenzierte Relation**.

Etwas lax kann die Fremdschlüsseldefinition auch so formuliert werden: Ein Fremdschlässel ist eine Art Kopie eines Primärschlüssels, jeder auftretende be-

[7] Da ein Primärschlüssel nur ein ausgewählter Schlüsselkandidat ist, also jeder Schlüsselkandidat dieselben wesentlichen Merkmale wie der Primärschlüssel hat, kann diese Definition problemlos auf Schlüsselkandidaten verallgemeinert werden.

stimmte Wert des Fremdschlüssels tritt auch als Primärschlüssel der referenzierten Relations*instanz* auf.

Das Konzept der Fremdschlüssel ist wesentliches Konzept bei der Transformation des E/R-Modells in ein Relationenmodell. Es schreibt offensichtlich, wie Beziehungen / Verbindungen zwischen verschiedenen Relationsschemata ausgedrückt werden können.

Ergänzung zur Notation: Fremdschlüssel

Die Notation erfolgt in Analogie des Punktes 2 bei Primärschlüsseln mit dem Kürzel

:FS auf <Schema-Name>.

Beispiel ergeben sich im folgenden Abschnitt.

3.2.2 Umformregeln

Das Ziel einer Transformation eines E/R-Entwurfs in einen Entwurf nach dem Relationenmodell ist in jedem Falle die vollständige Umsetzung der Semantik. Mit Blick auf eine später mögliche weiteren Umformung des Modells in einen Datenbankentwurf sollen Merkmale aus dem E/R-Modell, die nicht direkt in das Relationenmodell eingehen, in Textform als Anlage gespeichert werden. Und als zweite, nicht explizit genannte Bedingung an eine Umformung gilt, dass die Zahl der entstehenden Relationsschemata möglichst gering bleibt.

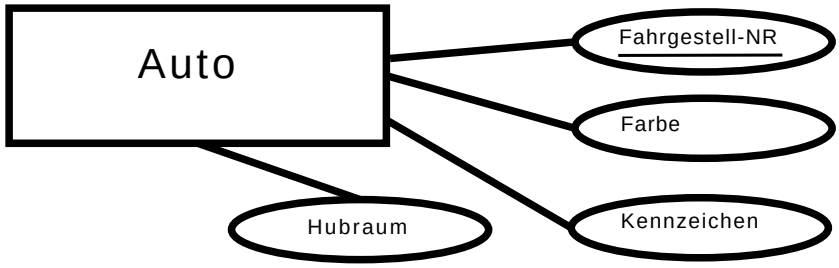
Attribute, starke Entitätsmengen

Regel 0: Die Beschreibung von einfachen Attributen aus dem E/R-Modell wird in das Relationenmodell übernommen. ■

Diese Regel bedarf keiner Erklärung.

Regel 1: Eine (starke) Entitätsmenge wird ein Relationenmenge $\mathcal{R}$. Jedes einfache Attribut wird übernommen. Mehrfache Attribute wie auch zusammengesetzte Attribute bilden jeweils eine eigene neue Relation, die über einen Fremdschlüssel mit $\mathcal{R}$ verbunden wird (Details s. Abschnitt 3.2.2, Seite 51). Der Primärschlüssel wird übernommen, sofern er aus nur aus einfachen Attributen gebildet wurde. Andernfalls ist er aus den Attributen neu zu bestimmen, die nach Auflösung der zusammengesetzten und mehrfachen Attribute verbleiben. Dabei kann es durchaus vorkommen, dass sich kein Schlüsselkandidat findet und ein künstlicher Primärschlüssel nach Abschnitt 2.3.2 (Seite 27) eingeführt werden muss. ■

Beispiel 3.12: Einfache Attribute
Ein Auto habe die folgenden Eigenschaften: Fahrgestellnummer, Farbe, Hubraum und ein Kennzeichen.



ER-Diagramm mit einfachen Attributen

Dieser Ausschnitt aus einem E/R-Schema wird zu folgendem Relationsschema:
Auto$(Fahrgestellnummer : PS, Farbe, Hubraum, Kennzeichen)$;
Die Kennzeichnung des Primärschlüssels ist selbsterklärend, eine formale Beschreibung erfolgt unter Punkt 4.2.

Relationship-Mengen

Regel 2: Eine Relationship-Menge bildet eine eigene Relation. Diese hat als Attribute jeweils einen Frendschlüssel auf jedes Relationsschema (das aus Entitätsmengen durch Anwendung der Regel 1 hervorgegangen ist) und auch jedes einwertige, nichtzusammengesetzte Attribut der Relationshipmenge wird als Attribut übernommen. Der Primärschlüssel der Relation umfasst mindestens alle Fremdschlüssel. Im Falle, dass die Relationschipmenge keine eigenen Attribute hat, ist der Primärschlüssel die Vereinigung aller Fremdschlüssel. ■

Regel 2a: Falls bei der Transformation einer Relationship-Menge eigene Attribute zu beachten sind, kann die Bildung des Primärschlüssels nicht formal erfolgen, sondern es müssen die für einen Primärschlüssel gültigen Bedingungen (siehe Definition 2.9:, Seite 17) durch Betrachtung der Semantik geprüft werden. Im Ergebnis dieser Prüfung werden dann noch weitere Attribute zur Vereinigung der Fremdschlüssel hinzugenommen. ■

Wie im Abschnitt 2.4.1 (Seite 29, vgl. auch 2.13:) bemerkt wurde, muss ein Fremdschlüssel gewissermaßen eine Kopie eines in einer (anderen oder derselben) Relation existierenden Primärschlüssels sein. dies drückt sich in dem Begriff referentielle Integrität aus. **R**

Definition 3.13: ***Referentielle Integrität*** *im Relationenmodell bedeutet, dass zu jedem Fremdschlüssel, der in einer Relation* **S** *verwendet wird, ein Primärschlüssel in einem Relationsschema* **R** *derart existieren muss, dass zu jedem Wert des Attributs , das den Fremd-*

schlüssel bildet und in einer Instanz von **S** *auftritt, eine Relation in der Instanz von* **R** *derart existieren muss, dass Wertgleichheit der Attribute aus* **S** *des Fremdschlüssels und des entsprechen Primärschlüssels aus* **R** *besteht.*

Die stark restriktive Bedeutung dieser Forderung nach referentieller Integrität in einem Relationenmodell besteht darin, dass diese Forderung nicht nur für die gerade in den Relationen existierenden Tupel gilt, sondern bei jeder Belegung der beteiligten Relationsschemata erfüllt sein muss ([8]). Die referentielle Integrität ist eine der **relationalen Invarianten**. Unter diesem Begriff werden einige der grundlegenden Eigenschaften des Relationenmodells zusammengefasst. Als Folgerung aus dieser Definition können die Werte von Attributen, die zu einem Fremdschlüssel gehören, nicht unbestimmt sein. Diese Restriktion wird in besimmten Umständen zu lockern sein (hierzu auch Abschnitte 3.2.2, S. 51 und 2.4.3, S. 31). Diese Fälle, in denen dann alle Attribute des Fremdschlüssels unbestimmt sind, sind dann keine Verletztung der referentiellen Integrität.

Beispiel 3.14: Relationship-Mengen vom Typ M:N

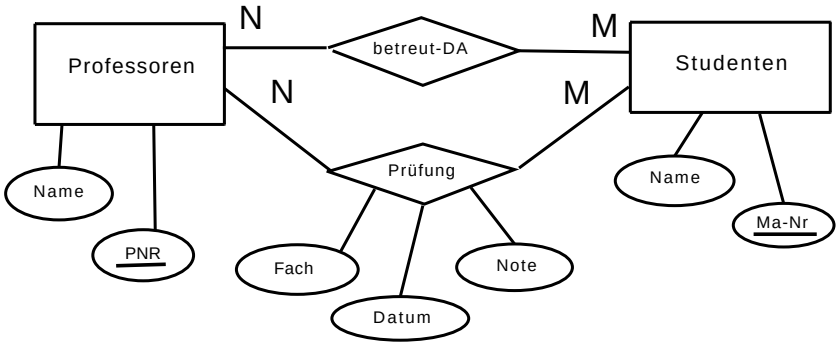


Abbildung 3.1: Typ M:N

Dazu gehört die Umsetzung:

Professoren$(PNR : PS, Name)$;

Studenten$(Ma - Nr : PS, Name)$;

betreut $-$ **DA**$(Prof, Stud), (Prof, Stud) : PS,$
$Prof : FS\ auf\ Professoren, Stud : FS\ auf\ Studenten$;

Prüfung$(Prof, Stud, Fach, Note, Datum),$
$(Prof, Stud, Fach, Datum) : PS,$
$Prof : FS\ auf\ Professoren, Stud : FS\ auf\ Studenten$;

[8] SQL bietet spezielle Konstrukte zur Pflege der referentiellen Integrität - es ist die Aufgabe des Datenbankentwicklers, diese Möglichkeiten zu nutzen.

dabei wurde unterstellt, dass ein Professor einen Studenten an einem Tag im gleichen Fach nur einmal prüft, was sicher eine sinnvolle Annahme ist. Dies ist notwendig, da ein Student bei einem Professor durchaus mehrere Prüfungen absolvieren kann, die Kombination der beiden Fremdschlüssel in diesem Falle keinen Schlüsselkandidaten liefert. Auch soll es eventuell mehrere Betreuer geben.
Dass die Namen der beiden Fremdschlüssel-Attribute Prof und Stud sind, hat keinen Einfluß auf ihre Semantik. Diese wird durch die Information über ihre Fremdschlüsseleigenschaft ausreichend erklärt.

Regel 2 ist allgemein gültig. Wenn die zu transformierende Relationship-Menge eine binäre Beziehung darstellt, die einen der Abbildungstypen 1:1, 1:N oder N:1 hat, dann wird das Ergebnis der Umformung vereinfacht und die Anzahl der Ergebnisrelationsschemata reduziert, indem das aus der Relationship-Menge entstandene Relationsschema sofort wieder aufgelöst wird und in eines der Schemata integriert wird, die aus den Entitätsmengen erhalten wurden, und zwar in dasjenige, aus dessen Sicht die Beziehung eindeutig ist.

Regel 3: Nur M:N Beziehungen und nichtbinäre Relationship-Mengen bilden eigene Relationsschemata. Die Abbildung von 1:N-Beziehungen (und 1:1-Beziehungen) geschieht unter Auflösung des Relationsschemas, welches aus der Relationshipmenge entstanden ist.

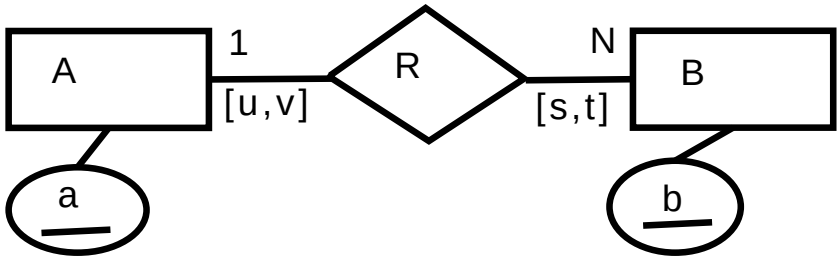


Abbildung 3.2: 1:N-Beziehung

In Abbildung 3.2.2 wird ausgesagt, dass die Beziehung **R** als Abbildung aus **B** in **A** eindeutig ist, dass also eine Entität aus **B** mit keiner oder höchstens mit eine Entität aus **A** in Beziehung steht. Anstatt nun aus **R** ein eigenes Relationsschemata zu erzeugen, kann auch in **B** ein zusätztliches Attribut a_1 eingeführt werden, welches einen Fremdschlüssel auf **A** realisiert:

$$\mathbf{A}(a : PS) \qquad \mathbf{B}(b : PS, a_1 : FS \text{ auf } A)$$

Der Vorteil dieses Vorgehens besteht darin, dass ein Relationsschema weniger entsteht und so später bei Anfragen weniger JOIN-Operationen zur Verknüpfung der Relationen benötigt werden, was i.A. eine Performanz-Steigerung bedeutet.

Bei 1:1-Abbildungen wird ebenso vorgegangen. Hier besteht hinsichtlich der Integration des neuen Fremdschlüssels die Möglichkeit, jede der Seiten auszuwählen. ■

Für Fremdschlüssel, die bei bei Auflösung von aus Relationshipmengen entstandenen Schemata bei 1:1- und 1:N-Beziehungen ([9]) entstehen, muss die im Abschnitt 3.2.2 (S. 48) geforderte Freiheit des Fremdschlüssels von unbestimmten Werten gelockert werden. Die Forderung nach Existenz der referenzierten Primärschlüssel wurde dort begründet. Die jetzt benutzten Fremdschlüssel sind in ihrer Form davon nicht zu unterscheiden, sie haben jedoch eine erweiterte Funktion. Sie müssen gegebenenfalls auch die zu der 1:1- oder 1:N-Beziehungen gehörigen Kardinalitätsrestriktionen (s. Abschnitt 2.4.3 (S. 31) ausdrücken können. Nimmt nun eine Entität, in derem Relationsschema die Beziehung als Fremdschlüssel realisiert wird, nicht an der Beziehung teil (weil eine Kardinalitätsrestriktion vom Typ [0,*] dies zulässt, kann der entsprechende Fremdschlüssel nicht besimmt werden. Formal wird dies durch unbestimmte Werte für die Attribute ausgedrückt, die zum Fremdschlüssel gehören und dann nicht als Verletzung der referentiellen Integrität bewertet.

Schwache Entitätsmengen, mehrfache und zusammengesetzte Attribute

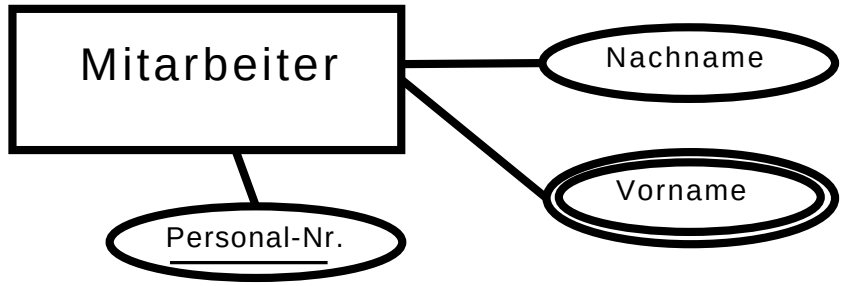
Wie sind schwache Entitätsmengen, mehrfache und zusammengesetzte Attribute zu transformieren? Die gemeinsame Idee für alle diese Fälle ist die Bildung neuer Relationsschemata, die über Fremdschlüssel angebunden werden.

Der Begriff „Mehrfaches Attribut“ bedeutet, dass es mehrere Werte eines Attributs zu einer Entität geben kann (vgl. Definition 2.5:, Seite 15), deshalb wird hier die folgende Regel definieret:

Regel 4: Zu mehrfachen Attributen wird ein eigenes Relationsschema mit einem neuen Namen erzeugt, dieses besteht aus einem Fremdschlüssel auf das Schema, zu dem das mehrfache Attribut gehört und dem Attribut. Als Primärschlüssel dient die Vereinigung aller Attribute. Die Beschreibung der mehrfachen Attribute geht in die Beschreibung der entstehenden Tabellen ein.

[9] Siehe „Abbildungstyp“ , S. 29.

Beispiel 3.15: Mehrfache Attribute



Mehrfache Attribute

Die Entitätsmenge liefert:

$$\textbf{Mitarbeiter}(\mathit{Personal-Nr.}:\mathit{PS},\mathit{Nachname});$$

Aus dem mehrfachen Attribut wird ein neues Relationsschema, dessen Name **Alle-Vornamen** *frei gewählt werden kann:*

$$\textbf{Alle-Vornamen}(\mathit{Personal-Nr.}:\mathit{FS\ auf\ Mitarbeiter},\mathit{Vorname}),$$
$$(\mathit{Personal-Nr.},\mathit{Vorname}):\mathit{PS};$$

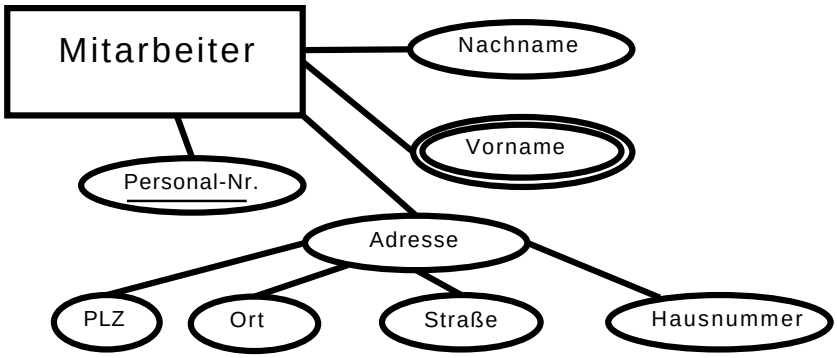
Der Fremdschlüssel stellt den Bezug zu dem konkreten Mitarbeiter her. Die Personalnummer allein reicht nicht, um den Primärschlüssel zu bilden, da entsprechend dem E/R-Schema ein Mitarbeiter mehrere Vornamen haben kann.Da jedoch ein Vorname bei einem Mitarbeiter nicht mehrfach auftritt, ist die Attributkombination (Personal-Nr.,Vorname) Schlüsselkandidat.

Zusammengesetzte Attribute beschreiben etwas in der Realität, das eine innere Struktur hat. Zusammengesetzte Attribute tragen im allgemeinen auch die Information, dass die Zusammensetzung eine eigene Semantik besitzt. Dies muss bei der Transformation gewahrt werden. Durch die Umformung, in der das zusammengesetzte Attribut zu einem neuen Relationsschema wird, wird dieses gewissermaßen als neue Einheit betrachtet. Der Bezug zur zu der ursprünglichen Entität, wo die Information als zusammengesetztes Attribut auftrat, wird wieder im neuen Relationsschema durch einen neu hinzugenommenen Fremdschlüssel. Wenn das Attribut nicht mehrfach ist, bildet der Fremdschlüssel auch den Primärschlüssel.

Regel 5: Zusammensetzte Attribute führen zu einem neuen Relationsschema, bestehend aus einem Freamdschlüssel auf das Schema, zu dem das zusammengesetzte Attribut gehört und den Attributen, die in der Zusammensetzung auftreten. Der Fremdschlüssel bildet gleichzeitig den Primärschlüssel.

Beispiel 3.16: Zusammengesetzte Attribute *Das vorangehende Beispiel wird um das Attribut* Adresse *erweitert, welches seinerseits aus PLZ, Ort, Straße, Hausnummer*

zusammengesetzt ist.



Die Relationsschemata **Mitarbeiter** *und* **Alle − Vornamen** *ergeben sich wie eben:*

$$\textbf{Mitarbeiter}(\mathit{Personal-Nr.}:\mathit{PS},\mathit{Nachname});$$
$$\textbf{Alle-Vornamen}(\mathit{Personal-Nr.}:\mathit{FS\ auf\ Mitarbeiter},\mathit{Vorname}),$$
$$(\mathit{Personal-Nr.},\mathit{Vorname}):\mathit{PS};$$

Aus dem Attribut Adresse wird ein Relationsschema **Adressdaten***:*

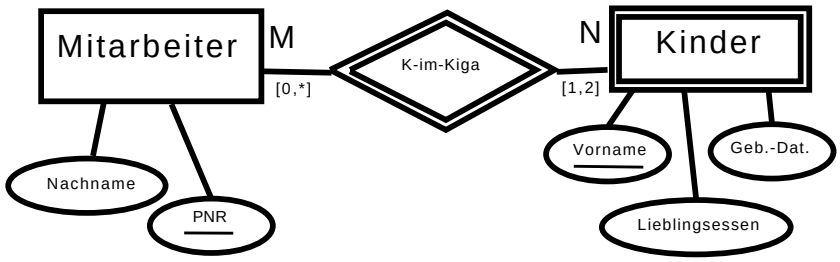
$$\textbf{Adressdaten}(\mathit{Personal-Nr.}:\mathit{PS}:\mathit{FS\ auf\ Mitarbeiter},\mathit{PLZ},\mathit{Ort},\mathit{Straße},\mathit{Hausnummer});$$

Der Fremdschlüssel stellt wieder den Bezug zu dem konkreten Mitarbeiter her. Der Primärschlüssel ist auch einfach, da jeder Mitarbeiter - wie im E/R-Diagramm modelliert - nur eine Adresse hat. Es bleibt dem Leser überlassen, die Aufgabe so zu erweitern, dass ein Mitarbeiter mehrere Adressen haben kann.

Es ist natürlich verlockend, statt der Bildung eines neuen Schema eine Integration der Attribute in das bestehende Schema vorzunehmen, die zusammengesetzte Struktur quasi „flachzuklopfen“. Ein solches Vorgehen stellt einen Fehler dar, da die Information über eigenständige Semantik der Zusammensetzung verloren geht.

Regel 6: Eine schwache Entitätsmenge wird in ein Relationsschema transformiert, deren Attribute aus einem Fremdschlüssel auf das Schema zur zugehörigen starken Entitätsmenge und den einfachen Attributen der schwachen Entitätsmenge bestehen. Der Primärschlüssel besteht mindestens aus dem Fremdschlüssel. Aus der Semantik der Attribute ist zu entscheiden, ob weitere Attribute der schwachen Entitätsmenge zum Primärschlüssel hinzuzufügen sind.

Beispiel 3.17: Schwache Entitätsmengen



Betriebskindergarten

Mitarbeiter$(PNR : PS, Nachname)$;
Kinder$(PNR, Vorname, GebDatum, Lieblingsessen)$,
$(PNR, Vorname) : PS, PNR : FS\ auf\ Mitarbeiter$;

Wie sieht es hier mit der Abbildung der Kardinalitätsrestriktionen aus? Auf der Seite der Mitarbeiter steht keine reale Einschränkung, dies wird auch korrekt abgebildet. Auf der Seite der Kinder wird verlangt, das mindestens ein Elternteil Mitarbeiter ist. Dies wird dadurch modelliert, dass das Attribut PNR bei Kindern Teil des Primärschlüssels nicht unbestimmt bleiben darf und als Fremdschlüssel, der nicht unbestimmt ist, einen Wert haben muss, der als Primärschlüssel in Mitarbeiter tatsächlich vorkommt. Ein Problem bereitet nur die obere Schranke 2 *in der Notation. Diese kann tatsächlich nicht im Relationsschema direkt abgebildet werden. Hier muss im Modell zu einer Anmerkung gegriffen werden, in der diese Forderung verbal notiert wird, damit sie in einer Datenbank beispielsweise durch einen Trigger realisiert werden kann.*

Regel 7: Bei komplexen Attributkonstruktionen müssen ggf. die Regeln1 bis 6 mehrfach angewendet oder kombiniert werden.

Umsetzung der Erweiterungen

Wie schon im Abschnitt 2.7 (Seite 35) angedeutet, wird eine Teilmengenbeziehungen zwischen Entitätsmengen im E/R-Modell nicht unterstützt. Somit ist zu beschreiben, wie die Vererbung nach Abschnitt 2.7 (Seite 35) in das Relationenmodell zu übertragen ist. Unter dem Oberbegriff „Partionierung“ werden drei Möglichkeiten zusammengefasst. Das Vorgehen soll durch folgendes Beispiel illustriert werden:

Vorgelegt sei die folgende Mitarbeiterliste, die sowohl Professoren als auch Angestelle der Universitätsverwaltung enthält. Letztere haben natürlich kein Forschungsgebiet.

Diese Personen sollen in Relationsschemata nach Abbildung 2.12 eingeordnet werden. Für die Lösung bieten sich mehrere Möglichkeiten an:

Alle − Mitarbeiter

PNR	Name	Vorname	Forschungsgebiet
2364	Rahm	Erhard	Datenbanken
0342	Beyer	Klaus	Analysis
9645	Wartenberg	Günther	Kirchengeschichte
4714	Cain	Hans-Ulrich	Klass. Archäologie
0815	Müller	Peter	
1230	Muster	Eva	

Tabelle 3.1: Beispieltabelle: Alle-Mitarbeiter

1. **Vertikale Partitionierung** Beide Entitätsmengen bilden jeweils ein Relationsschema, die Untermenge (hier: Professoren) wird so behandelt als wäre sie eine schwache Entitätsmenge, d.h. sie übernimmt als zusätzliches Attribut den Primärschlüssel der Obermenge auf, der in der Untermenge zugleich Primärschlüssel und Fremdschlüssel auf die Obermenge ist.

   Beispiel 3.18: *Es ergeben sich die Relationsschemata:*
   Mitarbeiter$(PNR : PS, Name, Vorname)$;
   Professoren$(PNR : PS : FS\ auf\ Mitarbeiter, Forschungsgebiet)$;

   mit den Belegungen:

   Mitarbeiter

PNR	*Name*	*Vorname*
2364	*Rahm*	*Erhard*
0342	*Beyer*	*Klaus*
9645	*Wartenberg*	*Günther*
4714	*Cain*	*Hans-Ulrich*
0815	*Müller*	*Peter*
1230	*Muster*	*Eva*

   Professoren

PNR	*Forschungsgebiet*
2364	*Datenbanken*
0342	*Analysis*
9645	*Kirchengeschichte*
4714	*Klass. Archäologie*

   Werden jetzt alle Merkmale eines Professors gesucht, sind die Informationen aus beiden Tabellen zusammenzubringen. In der Anfragesprache SQL erfodert dies eine JOIN-Operation.

2. **Horizontale Partitionierung** Diese Form könnte unter der Kurzform beschrieben werden: jede Relation steht in ihrer natürlichen Klasse. Jedes Schema enthält jeweils alle Angaben zu einer Person. Es werden wieder zwei Schemata gebildet, die Attribute entsprechen denen des E/R-Schema unter Beachtung der Vererbung.

   Mitarbeiter$(PNR : PS, Name, Vorname)$;
   Professoren$(PNR : PS, Name, Vorname, Forschungsgebiet)$;

Mitarbeiter

PNR	Name	Vorname
0815	Müller	Peter
1230	Muster	Eva

Professoren

PNR	Name	Vorname	Forschungsgebiet
2364	Rahm	Erhard	Datenbanken
0342	Beyer	Klaus	Analysis
9645	Wartenberg	Günther	Kirchengeschichte
4714	Cain	Hans-Ulrich	Klass. Archäologie

Da jetzt die Mitarbeiter über mehrere Tabellen verteilt sind, sind bei Aussagen über alle Mitarbeiter die Informationen aus beiden Tabellen zusammenzubringen.

3. **Partitionierung mit voller Redundanz** Diese Methode hält Daten redundant vor, was einige Anfragen vereinfacht. Als Preis dafür steht erhöhter Speicherbedarf und erhöhter Aufwand bei der Wahrung der Konsistenz der Daten.
Die Relationsschemata sind wie bei der horizontalen Partitionierung definiert.
Mitarbeiter$(PNR : PS, Name, Vorname)$;
Professoren$(PNR : PS, Name, Vorname, Forschungsgebiet)$;

   Aktuelle Belegung: Daten von Professoren sind zum Teil redundant gespeichert.

Mitarbeiter

PNR	Name	Vorname
0815	Müller	Peter
1230	Muster	Eva
2364	Rahm	Erhard
0342	Beyer	Klaus
9645	Wartenberg	Günther
4714	Cain	Hans-Ulrich

Professoren

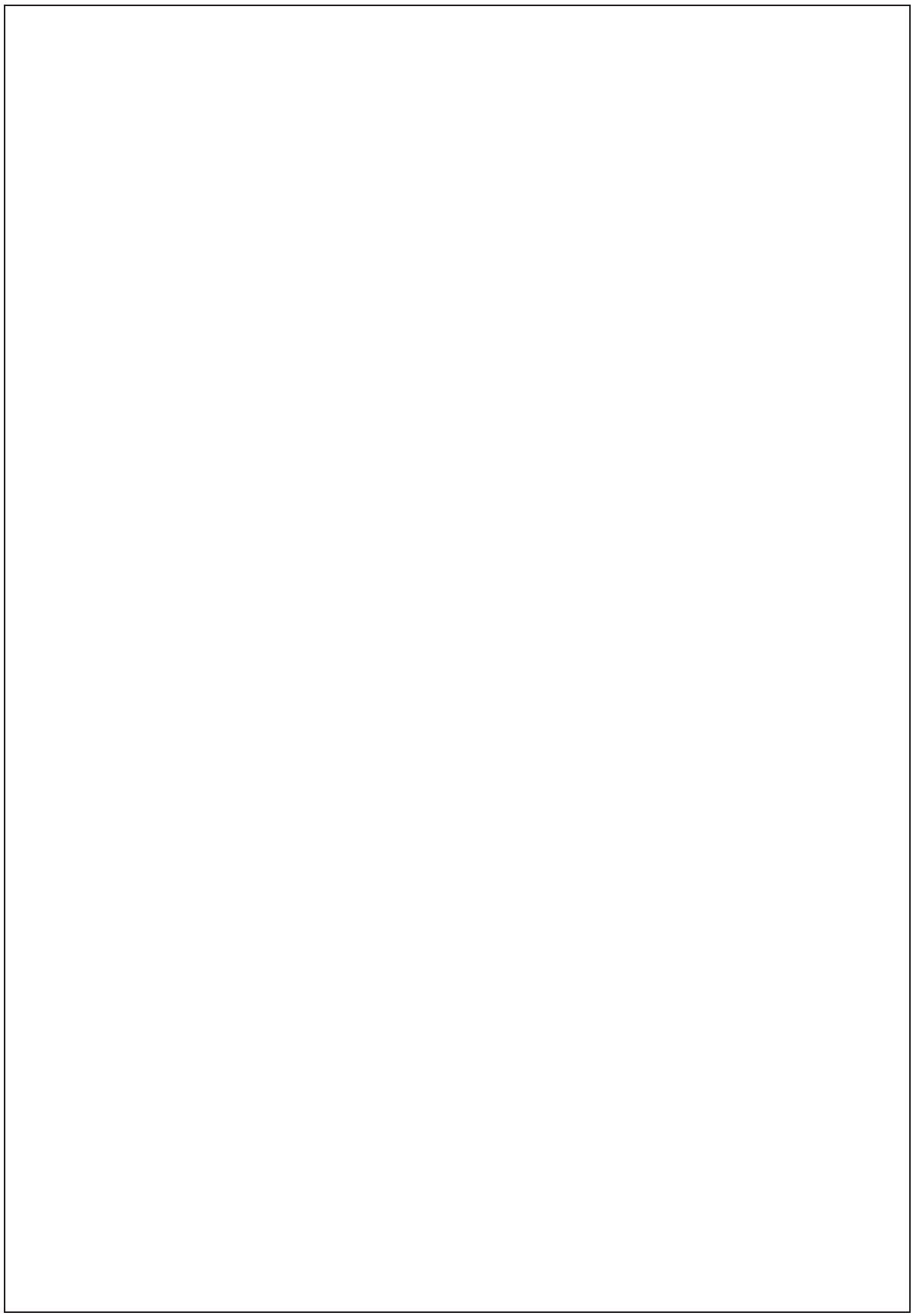
PNR	Name	Vorname	Forschungsgebiet
2364	Rahm	Erhard	Datenbanken
0342	Beyer	Klaus	Analysis
9645	Wartenberg	Günther	Kirchengeschichte
4714	Cain	Hans-Ulrich	Klass. Archäologie

   Bei dieser Darstellung kann die Information stets aus einer Tabelle erhalten

werden, jedoch erfordert die Sicherung der Konsistenz beider Tabellen bei Insert- und Update-Operationen zusätzlichen Auswand.

3.3 Beispiel

Der E/R-Entwurf, den Sie auf Seite 38 zur Banken-Miniwelt aus Abschnitt 2.24: erarbeitet haben, soll in Relationsschemata übertragen werden, so dass alle semantischen Merkmale möglichst vollständig in dem neuen Entwurf dargestellt werden und die Zahl der verwendeten Schemata möglichst gering ist.



3.4 Rücktransformation in das E/R-Modell

In seltenen Fällen kann die Aufgabe entstehen, aus dem Relationenmodell ein E/R-Modell abzuleiten, um zum Beispiel bei Datenintegrationsaufgaben die semantischen Zusammenhänge besser zu verstehen.
Zunächst die Festestellung:

Definition 3.19: *Ein E/R-Schema gilt als Ergebnis einer Rücktransformation von Relationsschemata, wenn aus dem E/R-Schema durch Anwendung der Regeln nach Abschnitt 3.2 (Seite 45) die Relationsschemata entstehen.*

Satz 3.20: *Die Transformation eines Relationenmodells in ein E/R-Schema ist i.a. nicht eindeutig.*

Zum Beweis dieser Negativaussage reicht die Angabe eines Beispiels, in dem aus zwei verschiedenen E/R-Schemata das gleiche Relationsschema gewonnen wird. Ein solches Beispiel wurde durch Aufgabe 2.14: (Seite 25) gegeben, die beiden durch die Abbildungen 2.1 und 2.2 aufgezeigten Lösungen ergaben bei Anwendung der Regeln das gleich Relationenmodell (siehe Beispiel 3.17:, Seite 54). ■

Wesentlich für die Rücktransformation ist das Erkennen von Entitäts- und Relationship-Mengen. Das ist letztlich ein Vorgang, der das Verstehen der Semantik erfordert. Dennoch gibt es formale Merkmale, die eine Zuordnung zumindest wahrscheinlich machen.

Wenn vorausgesetzt wird, dass die Fremdschlüsselbeziehungen auf die Primärschlüssel verweisen (und nicht auf andere Schlüsselkandidaten - vgl. Fußnote auf Seite 46), lassen sich folgende Aussagen angeben:

1. In einer Relation gibt es keine Fremdschlüssel. Die Elemente stellen wahrscheinlich Entitäten dar, die Relation entspricht einer Entitätsmenge
2. In einer Relation existiert ein zusammengesetzter Primärschlüssel, Teile der Bestandteile des Primärschlüssels sind gleichzeitig die Fremdschlüssel auf schon erkannte Entitätsmengen. Diese Relation gehört wahrscheinlich zu einer Relationship-Menge, die, wenn sie binär ist, den Abbildungstyp M:N hat. Alle Attribute, die nicht den Fremdschlüsseln angehören, sind eigene Attribute der Relationship-Menge.
3. Eine Relationen, deren Primärschlüssel gleichzeitig Fremdschlüssel auf genau eine Relation ist, die schon als Entitätsmenge erkannt wurde,
    - repäsentiert möglicherweise ein zusammengesetztes, einwertiges Attribut der durch den Fremdschlüssel referenzierten Entitätsmenge. oder

- ist das Ergebnis einer aufgelösten (nach Beispiel) 1:1-Beziehung. Diese Konstellation tritt typischerweise auf, wenn die Semantik der Beziehung eine Teilmengenbeziehung erkennen lässt.

4. Eine Relation besitzt einen zusammengesetzten Primärschlüssel, von dem ein Teil gleichzeitig Fremdschlüssel auf eine schon erkannte Entitätsmenge ist. Zu dieser Situation gehört das oben gegebene Beispiel der nicht eindeutigen Rücktransformation. Es gibt zwei Varianten, deren Semantik sich unterscheidet. Hier helfen formale Merkmale nicht weiter. Hier muss i.A. ein Mensch zur Entscheidung eingreifen:
   - Die beschriebene Relation ist eine schwache Entitätsmenge mit bezug auf diejenige Entitätsmenge, die durch den Fremdschlüssel referenziert wird. Dann gehören die Attribute, die den Fremschlüssel bilden, nicht ursprünglich zur schwachen Entitätsmenge, sondern sie sind im Prozess der Schlüsselbildung aus der referenzierten Entitätsmenge übernommen worden. Alle Attribute, die nicht zum Fremdschlüssel gehören, sind Attribute der schwachen Entitätsmenge, diejenigen, die davon zum Primärschlüssel der Relation gehören, bilden die Schlüsselergänzung.
   - Die beschriebene Relation ist ein mengenwertiges, zusammengesetztes Attribut derjenigen Entitätsmenge, die Attribute, die nicht zum Fremschlüssel gehören, bilden die Komponenten der Zusammensetzung.
5. Ein Fremdschlüssel in einer Relation, die schon als Entitätsmenge erkannt wurde und dessen Attribute disjunkt zu den Schlüsselattributen sind, weist auf eine 1:1- oder 1:N-Beziehung hin. Dabei ist die Beziehung aus Sicht dieser Relation eindeutig.

Mit dieser Aufzählung von Umformungsregeln wurde ein Grundgerüst geliefert. Es ist mit hinreichender Vorsicht einzusetzen, da es grundsätzlich problematisch ist, aus formalen Merkmalen auf die Semantik zu schließen. Auch sind die angegebenen Regeln nicht vollständig, z.B. hinsichtlich einer mehrfachen Kombination von mehrfachen und zusammengesetzten Attributen. Auch werden 1:1-Abbildungen durch die Regeln nur als Typ 1:N bzw. N:1 klassifiziert. Auch die formale Erkennung der Kardinalitätsrestriktionen bleibt i.a. offen. Eventuell läßt sich aus der Zugehörigkeit von Attributen zu einem Fremdschlüssel über die Forderung der referenziellen Integrität schließen, dass die Werte vorhanden sein müssen.

3.5 Bewertung des Relationenmodells

Das Relationenmodell ist stark auf die Realisierung der Schemata in Form einer relationalen Datenbank ausgerichtet. Es entspricht etwas dem Niveau, welches SQL mit dem Stand von 1992 erreicht hatte. Somit kann Definitionsteil von SQL:1992 als Möglichkeit der formale Beschreibung eines Relationenmodells betrachtet werden. Ein erfahrener Datenbankentwickler wird mit Hilfe der hier beschriebenen

Umformregeln einen E/R-Entwurf direkt in die entsprechenden SQL-Befehle zur Tabellendefinition umsetzen. Vielfach ist ein semiformaler Entwurf mit Kommentaren, wie er in den vorangehenden Abschnitten benutzt wurde, geeignet, eine Übersicht zu erlangen. Damit wird die Kenntnis des Relationenmodells eine wichtige Voraussetzung für einen effizienten und korrekten Entwurf einer relationalen Datenbank.
Als Mangel ist anzusehen, dass keine Beschreibung der Semantik vorgesehen ist([10]).
Wird eine Form des Relationenmodells gesucht, die formal beschrieben bzw. durch eine Maschine verarbeitbar ist, ist der RÃ¼ckgriff auf die in SQL enthaltene Datendefinitionssprache (DDL: Data Definition Language) angezeigt.

Mit dem Relationenmodell wird der mathematische Begriff der Algebra in den Datenbankentwurf eingebracht. Dies ermöglicht, eine Relationenalgebra zu definieren, die eine theoretische Grundlage der Datenabfragesprache darstellt und die Grundlage der Anfragezerlegung und Anfrageoptimierung in einen relationalen Datenbankverwaltungssystem darstellt.

[10] Dieser Mangel wird leider in SQL nicht behoben, es bleibt in der Verantwortung des Entwicklers, dies wenigstens in Form von informellen Kommentaren und Dokumentationen teilweise zu kompensieren.

4 Anhang

4.1 Bankenbeispiel als E/R-Schema

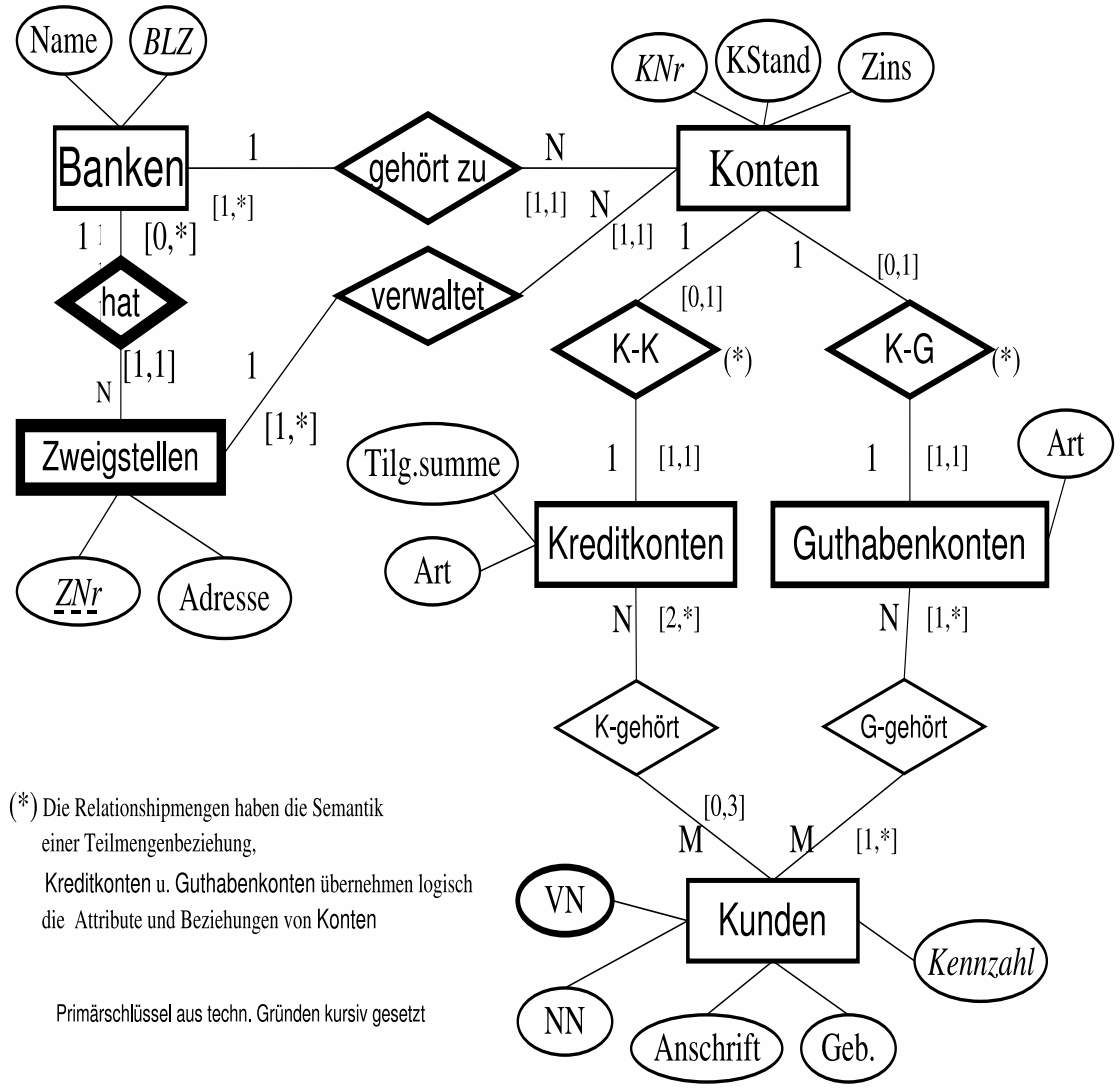


Abbildung 4.1: Lösungsvorschlag - ohne UML-Elemente

Kommentare:
Die vorgestellte Lösung bedarf einiger Kommentare. Diese zeigen, wie sehr ein E/R-Entwurf vom Verständnis des in der Miniwelt beschriebenen Sachverhalts abhängt.
Unstrittig ist sicher das die Banken eine Entitätsmenge *Bank* bilden. Schon die Festlegung, dass die Bankleitzahl setzt Weltwissen oder das Erkennen vor aus, dass eine *Leitzahl* hier charakteristisches Attribut ist. Aus der Eigenschaft, dass Zweigstellen Konten verwalten, kann geschlossen werden, dass sie eine gewisse Bedeutung im Modell haben. Über Zweigstellen sind zwei Fakten bekannt, eine Adresse und eine innerhalb der Bank eindeutige Zweigstellennummer. Denkt man an große Städte mit großen Gebäuden, wird schnell klar, dass sich in einem Gebäude (z.B. in einem Einkaufspark) durchaus mehrere Zweigstellen verschiedener Banken befinden können. Auch die Zweigstellennummer ist kein Schlüsselkandidat, da sie nur bezüglich jeder Bank eindeutig ist. Deshalb ist es (theoretisch) möglich, dass sch an einer Adresse zwei Zweigstellen zweier Banken befinden, die zufällig auch die gleichen Zweigstellennummern haben. Auf Grund dieser Überlegungen werden Zweigstellen als schwache Entitätsmengen mit Bezug zu *Bank* modelliert.
Ein Konto ist natürlich existenzabhängig von der zugehörigen Bank, trotzdem ist hier die schwache Entitätsmenge nicht das Mittel zur Modellierung. Wegen des Satzes „Jedes Konto hat eine innerhalb der Miniwelt eindeutige Kontonummer " findet sich die Kontonummer als Schlüsselkandidat, damit wird Konto zu einer starken Entitätsmenge.
Zwischen der Entitätsmenge *Konto* und den Arten *Guthaben- und Kreditkonten* bestehen offensichtlich Teilmengenbeziehungen, die natürlich auch durch die der UML entlehnten Symbolik modelliert werden könnte. Es wäre hier weniger gut, die Kontoart durch ein Attribut zu beschreiben, da die beiden Arten unterschiedliche Attribute tragen. Dieses Argument gilt für die Unterarten nicht mehr, deshalb werden sie durch Attribute modelliert. Zu beachten ist dabei, dass die Attribute *Art* bei Kredit- und Guthabenkonto jeweils einen unterschiedlich Wertevorrat haben. Die mehrfache Verwendung des Attributnamens *Art* ist hier kein Problem, da die beiden Arten durch den Kontext (nämlich durch ihre Zugehörigkeit zu unterschiedlichen Entitätsmengen) unterschieden werden.
Die Relationenschemata mit den Namen *ist ein* stellen vereinbarungsgemäß eine Besonderheit dar. Zum einen werden hier zwei solche Schemata mit dem gleichen Namen belegt, was i.a. unzulässig ist. Und zum anderen wird bei diesem Namen unterstellt, dass das Schema der Untermenge alle Attribute besitzt, die die Obermenge auch hat, also *Kreditkonto* hat auch die Attribute *KNr, KStand, Zins* hat. Nur mit der Kenntnis dieser Vereinbarung ist die Lösung einsichtig, dann aber sehr übersichtlich.

4.2 Notation

Zusammenfassend werden hier nochmal die Festlegungen zur semiformalen Notation von Relationenschema aufgeführt, wie sie im Text im Relationenmodell benutzt worden:

```
att-list
    ::=   <att-name> :PS[, att-rest-list ]
        | <att-name> :PS :FS auf <schema-name>[, att-rest-list ]

att-name-list
    ::=   <att-name>
        | <att-name>, att-name-list

att-rest-list
    ::=   <att-name>[ :FS auf <schema-name> ]
        | <att-name>[ :FS auf <schema-name> ], att-rest-list
fs-ausdruck
    ::=   (att-name-list) :FS auf <schema-name>
        | (att-name-list) :FS auf <schema-name> -E
          (Bem.: -E = der referenzierte Wert muss nicht existieren)

fs-list
    ::=   fs-ausdruck
        | fs-ausdruck, fs-list

key-vek
    ::=   (att-name-list) :PS [, fs-list ]
        | fs-list

rel-schema
    ::=   <schema-name>( att-name-list ), key-vek [, rem ];
        | <schema-name>( att-list )[, rem ];

rem
    ::=   % <beliebiger Text>                                  Kommentar
```

Bei diesen Festlegungen wurde beachtet, dass es möglich sein soll, einfache Schlüssel direkt bei den Attributen zu erklären, zusammengesetzte Schlüssel werden stets nach der Attributnamensliste angegeben.
Mit dem Feld **rem** sollen alle Bemerkungen ermöglicht werden, die in dieser vereinfachten Notation sonst nicht darstellbar wären: Pflichtattribute, Ausschluss von Duplikaten, ... , aber auch die Beschreibung der Semantik.

Soll die Definition 2.2: , Seite 13 für Attribute umgesetzt werden, bieten sich folgende Regeln an:

attribut
::= Attribut <attribut-name>(wertevorrat, context, semantik);
| Attribut <attribut-name>
~ Attribut [<schema-name>:]<attributname>(context
[, semantik]);
(Bem.: Semantisch gleiche Attribute - s.S. 14,
Ergänzungen zur Semantik möglich)

wertevorrat
::= Wertevorrat={ <Beschreibung des Wertevorrats> }

context
::= Context={ schema-name-list }
(Bem.: Liste aller Schemata, in denen das Attribut
verwendet wird.)

schema-name-list
::= <schema-name> [, schema-name-list]

semantik
::= Semantik={ <Beschreibung der Semantik> }

4.3 Bankenbeispiel - Relationenmodell

Relationen, die aus Entitätsmengen entstanden sind:

Banken

Attribut BLZ (
Wertevorrat={ Folge von maximal 14 Ziffern},
Context={ Banken, Zweigstellen},
Semantik={ in Deutschland eindeutige Numerierung der Banken}
);
Attribut Name (
Wertevorrat={ Folge von Zeichen },
Context={ Banken },
Semantik={ Name der Bank, evt. unter einem Namen Banken in verschiedenen Ländern, die zwar zusammengehören, aber wirtschaftlich selbständig agieren.}
);

Banken(BLZ :PS, Name);

Zweigstellen

Attribut ZNr (
Wertevorrat={ natürliche Zahl },
Context={ Zweigstellen, Konten},
Semantik={ Lfde Nummer einer Zweigstelle, beginnt für jede Bank neu bei 1 }
);
Attribut Adresse (
Wertevorrat={ Folge von Buchstaben und Zahlen },
Context={ Zweigstellen},
Semantik={ Postanschrift oder Hausanschrift der Zweigstelle }
);

Zweigstellen(ZNr, Adresse, BLZ :FS auf Bank), PS:(BLZ,ZNr),
% schwache E-Menge bezügl. Bank, deshalb BLZ) ;

Kunden

Attribut Kennzahl (
Wertevorrat={ Folge von Buchstaben und Zahlen },
Context={ Kunden, K-gehoert, G-gehoert},
Semantik={ eindeutige Identifikation einer Person (Kunde) , kuenstl. PS }
);

Attribut Vorname (
Wertevorrat={ Folge von Buchstaben, mehrere Wörter },
Context={ Kunden},
Semantik={ Vorname der Person}
);

Attribut Name (
Wertevorrat={ Folge von Buchstaben },
Context={ Kunden},
Semantik={ Nachname der Person}
);

Attribut Geb-Datum (
Wertevorrat={ Datum nach DIN-Norm Tag.Monat.Jahr XX.MM.JJJJ, nur zulässige Werte. },
Context={ Kunden},
Semantik={ Datum der Geburt der Person}
);

Attribut Anschrift (
Wertevorrat={ Zeichenfolge },
Context={ Kunden},
Semantik={ Anschrift, nicht genauer strukturiert}
);

Kunden(Kennzahl :PS, Vorname, Name, Geb-Datum, Anschrift);

Konten

Attribut KNr (
Wertevorrat={ Ziffernfolge},
Context={ Konten, Guthabenkonten, Kreditkonten },
Semantik={ identifiziert ein Konto }
);

Attribut KStand (
Wertevorrat={ Festkommazahlen mit zwei Nachkommastellen },
Context={ Konten },
Semantik={ Geldmenge auf dem Konto in Euro }
);

Durch

Attribut Zinssatz (
Wertevorrat={reelle Zahl z : $0 \leq z \leq 100$},
Kontext={ Konten },
Semantik={ Zinssatz pro Jahr}
);

Auflösung von Relationship-Mengen kommen folgende Attribute hinzu:

Attribut gehört-zu $\sim$ **Attribut** BLZ (
Context={ Konten },
Semantik={ Verweis auf kontoführende Bank}
);

Attribut wird-verwaltet $\sim$ **Attribut** ZNr (
Context={ Konten },
Semantik={ Verweis auf kontoführende Zweigstelle}
);

Konten(KNr :PS , KStand, Zinssatz, gehört-zu :FS auf Bank,
wird-verwaltet-von: FS auf Zweigstelle);

Kredit- und Guthabenkonten
Diese haben bezüglich der Relation „Konten“ jeweils die Semantik einer Teilmengenbeziehung (is-a-Semantik). Die Übertragung ins Relationenmodell soll nach dem Methode der vertikalen Partitionierung (vgl. Abschnitt 3.2.2, Seite 54)erfolgen. Das Attribut„Art“ ist in Guthaben- und in Kreditkonten unterschiedlich genutzt, dies ist an den unterschiedlichen Wertevorräten zu erkennen. Die Unterschiede verhindern, dass „Art“ zu einem Attribut des Oberbegriffs „Konto“ wird.

Attribut Art (
Wertevorrat={„Girokonto“, „Sparkonto“},
Context={ Guthabenkonten },
Semantik={ verschiedene Arten haben verschiedene Möglichkeiten }
);

Guthabenkonten(KNr :PS :FS auf Konto, Art);

Attribut Art (
Wertevorrat={„Baukredit“, „Allg.Kredit“},
Context={ Kreditkonten },
Semantik={ verschiedene Arten haben verschiedene Möglichkeiten }
);
Attribut Tilgungssumme (
Wertevorrat={Festkommazahlen mit zwei Nachkommastellen},
Context={ Kreditkonten},
Semantik={ monatlicher Betrag, der zur Tilgung vom Schuldner einzuzahlen ist. }
);

Kreditkonten(KNr :PS :FS auf Konto, Tilg.summe, Art);

% Kreditkonten und Guthabenkonten sind disjunkt -
% nicht im Relationenmodell darstellbar.;

Der letzte Kommentar ist von semantischer Relevanz, er darf nicht fehlen, denn bei einer Umsetzung in SQL-Befehle zum Erzeugen einer Datenbank kann diese Bedingung zum Beispiel mit Triggern realisiert werden.

Relationen aus M:N-Beziehungen:

Attribut G-K-Nr ~ **Attribut** KNr(
Context={ G-gehört },
Semantik={ FS auf Guthabenkonto}
);
Attribut Kunde ~ **Attribut** Kennzahl(
Context={ G-gehört }
);

G-gehört(G-K-Nr :FS auf Kreditkonten, Kunde :FS auf Kunden),
% Jeder Kunde muss mind. einmal in der Relation G-gehört als FS auftreten;

AttributK-K-Nr ∼ **Attribut** KNr(
Context={ K-gehört },
Semantik={ FS auf Kreditkonto}
);
Attribut Kunde-1 ∼ **Attribut** Kennzahl(
Context={ K-gehört }
);
Attribut Kunde-2 ∼ **Attribut** Kennzahl(
Context={ K-gehört }
);

K-gehört(K-K-Nr :FS auf Kreditkonten,
Kunde-1 :FS auf Kunden,
Kunde-2 :FS auf Kunden,
% Kunde-1 ≠ Kunde-2),

% freie Auslegung der Umformregeln, um zu erzwingen, dass jedes
% Kreditkonto mindestens zwei Schuldner hat;

4.4 Eigene Notizen

Literaturverzeichnis

[1] Chen, P.: *The entity-relationship model: Toward a unified view of data*. ACM Transactions on Database Systems, 1(1):9–36, 1976.

[2] Codd, E.: *A Relational Model for Large Shared Data Banks*. CACM, 13:6, Juni 1970.

[3] Elmasri, E. und Navathe, R.E.: *Grundlagen von Datenbanksystemen*. 3.Auflage,dt.,Pearson Studium, München, 2002.
ISBN 3-8273-7021-3

[4] Klaua, D.: *Allgemeine Mengenlehre*. Akademie-Verlag, Berlin, 1964.

[5] Lausen, G. und Vossen, G.: *Objektorientierte Datenbanken: Modelle und Sprachen*. R. Oldenburg Verlag München, 1996.

[6] Naas, J. und Schmid, H.L. (Herausgeber): *Mathematisches Wörterbuch, Bd. 1 und 2*. Akademieverlag Berlin und B.G. Teubnerverlagsgesellschaft Leipzig, 1961.

[7] Rahm, E.: *Datenbanksysteme*.
Vorlesung an der Universität Leipzig.
URL: http://dbs.uni-leipzig.de/en/lehre/db-lernmaterial-vorl.html
Meist jährliche Überarbeitung

[8] Weizenbaum, J.: *Computer Power and Human Reason. From Judgement to Calculation.* W. H. Freeman and Company. 1976
Deutsch als *Die Macht der Computer und die Ohnmacht der Vernunft*, Suhrkamp, Frankfurt am Main 1977, ISBN 3518278746

Index

(min,max)-Notation, 31
1:1, 50
1:1-Abbildung, 29
1:N, 50
1:N-Abbildung, 29

Abbildung
    1:1, 51
Abbildungstyp, 29
    1:1, 29
    1:N, 29
    N:M, 30
Abrial
    Jean-Raymond, 32
Algebra, 7
Anhang, 63
    Bankenbeispiel, E/R, 63
    Eigene Notizen, 72
    Lösung, Relationenmodell, 67
Attribut, 13
    einfaches, 15, 47
    Entitätsmenge, 17
    Gleichheit, 14
    mehrfaches, 15, 52
    Symbol, 20
    Transformation, 47, 51
    Unbestimmte Werte, 18
    zusammengesetztes, 15, 52
Attributmenge, 43

Beispiel, 22
    Abbildungstyp, 31
    Attribut, 14
    Banken-Miniwelt, 37
    einfaches Attribut, 47
    Entitätsmenge, 22
    M:N, 49
    mehrfache Attribute, 52
    Partitionierung, 55
    Relationsschema, 41
    schwache Entitätsmenge, 54
    zusammengesetzte Attribute, 52
Beziehung, 22
    binäre, 28
    is-a-, 35
    n-stellig, 33
    reflexive, 24
binär, 28, 29

Chen, 3
Codd, 3, 39
    Ted, 3

Datenintegration, 46
Definition
    Abbildungstyp, 29
    Attribut, 13
      einfaches, 15
      mehrfaches, 15
      zusammengesetztes, 15
    Entität, 13
    Entitätsmenge, 16
      Beschreibung, 16
    Fremdschlüssel, 46
    Gleichheit
      Attribute, 14
    Kardinalitätsrestriktion, 33
    Relation, 40, 43
    Relationsschema, 41
      Äquivalenz, 43
    Schlüsselkandidat, 17
    Schwache Entitätsmenge, 25

E/R-Modell, 3, 13
    Abbildungstyp, 29
    Attribute, 13

Bewertung, 34
Defizite, 34
Entitätsmengen, 13
Graphische Darstellung, 19
Kardinalitätsrestriktion, 31
Primärschlüssel, 17
Realtionship-Menge, 22
Rolle, 24
Rollen, 23
Schlüsselkandidat, 17
Schwache Entitätsmengen, 25
Unbestimmte Werte, 18
Entität, 13
Entität,Beschreibung im E/R-Modell, 15
Entitätsmenge, 16
schwache, 25, 54
Symbol, 20

Fremdschlüssel, 46
unbestimmte Werte, 51

Gegenstandsintegrität, 19
Grad, 28, *siehe* Relation
Relationship-Menge, 28

Hollerith, 9

Informatik, 9
Integrität
Gegenstands-, 19
referentielle, 48, 60
Invariante
relationale, 49
is-a, 35, 54

Kardinalität, 43
Kardinalitätsrestriktion, 31
1:1-,1:N-Abbildung, 32
is-a Beziehung, 35
n-stellige Beziehung, 33
Komponenten, 37
Komponentenbeziehung, 37
Kontext
Attribut, 14
Entitätsmenge, 17
Kreuzprodukt, 23, 40

Lösung
Bankenbeispiel, E/R, 63
Relationenmodell, 67

M:N, 49
M:N-Abbildung, 30
Menge
Attribute, 43
Entitäts-, 16
Relationship-, 22
Metasprache, 14
min,max-Notation
Einschränkung, 32
Miniwelt, 7, 13
Modell, 7
Besonderheiten der Informatiik, 9
Definition, 7
E/R-, 3, 13
hierarchisches, 3, 39
Informatik, 10
Korrektheit, 8
Netzwerk-, 3, 39
Relationen-, 3, 39
UML-, 3
Modellieren, 7, 8
Datenbank, 11

n-är, 28
N:1, 50
Name, *siehe* Relation
Attribut, 13
Entitätsmenge, 16
Normalform
-enlehre, 40
erste, 40
Notation, 42
Fremdschlüssel, 47, 65
Primärschlüssel, 65
semiformale, 65
SQL, 42

obligatorische Teilnahme, 32

optionale Teilnahme, 32

Partitionierung, 54
    horizontale, 55
    redundante, 56
    vertikale, 55
Primärschlüssel, 18, 36, 44
    künstlicher, 27
    Schlüsselergänzung, 25
    Symbol, 20

Regel 0, 47
Regel 1, 47
Regel 2, 48
Regel 2a, 48
Regel 3, 50
Regel 4, 51
Regel 5, 52
Regel 6, 53
Regel 7, 54
Reihenfolge
    Attribute, 40, 43
    Tupel, 44
Relation, 40, 43
    Grad, 41, 43
    Kardinalität, 43
    Name, 43
    referenzierende, 46
    referenzierte, 46
relationale Invariante, 49
Relationenalgebra, 7
Relationenmodell, 39
Relationschipmenge
    Relationsschemata, 59
Relationship-Menge, 22, 48
Relationship-Menge, Grad, 28
Relationshipmenge
    reflexiv, 24
    Symbol, 23
Relationsinstanz, 44
Relationsschema, 41
    Äquivalenz, 43
Relationszustand, 44
Rolle, 23, 24

Russel
    R.C., 9

Schlüssel, *siehe* Fremdschlüssel, *siehe* Primärschlüssel, *siehe* Schlüsselkandidat
Schlüsselkandidat, 17, 44
    Auswahl, 18
    Existenz, 18
    kein S., 27
    künstlicher, 27
    Minimalität, 18
Schwache Entitätsmengen, 25
Semantik
    Attribut, 14
    Entitätsmenge, 17
    schwache E., 26
Shannon, 9
Soundex, 9
SQL, 42
Steinbruch
    Karl, 9
Stern, 32
Symbol, 19
    Abbildungstyp, 29
    Attribut, 20
      mehrfach, 20
      zusammengesetzt, 20
    Entitätsmenge, 20
    Kardinalitätsrestriktion, 31
    Primärschlüssel, 20
    Relationshipmenge, 23
    schwache E., 25

Teilmenge, 35, 54
Teilmengenbeziehung, 36
Teilnahme
    obligatorisch , 32
    optionale, 32
Transformation
    E/R-Schema, 45
    Entitätsmenge, 47
    mehrfaches Attribut, 51
    Relationship-Menge, 48

Relationsschemata, 59
schwache Entitätsmenge, 51
zusammengesetztes Attribut, 51
Tupel
Reihenfolge, 44

UML, 35
Unbestimmte Werte, 18

Vererbung, 35, 54
Vorwort , 3

Weizenbaum, 10
Wert
unbestimmter, 18
Wertebereich
Attribut, 14